CL-001

誤り訂正符号に基づく偽装 QR コードの構成法とその脅威
A Construction of Fake QR Codes Based on Error-Correcting Codes and Its Threat

瀧田 愼[1)]　大熊 浩也[1)]　森井 昌克[1)]
Makoto Takita　Hiroya Okuma　Masakatu Morii

1　はじめに

二次元コードの一種である Quick Response(QR) コード [1][2] は，スマートフォンに搭載されたカメラで容易に情報を読み取れるため情報の伝達手段として幅広く使用されている．QR コードは部品・製品管理や流通経路の管理などの産業分野での利用を想定して開発された２次元コードである．産業分野での運用にも耐えうるように，製品の保管中や運搬中に QR コードの一部が破損したりよごれたりしても正確な情報を読み出せるように，情報の自己修復機能を備えている．開発当初は格納情報を取り出すために専用の QR コードデコーダが必要であり用途が限られていた．携帯電話やスマーフォンに QR コードデコーダが標準で，あるいはアプリで実装されたことから一般に広まっている．その利便性から，ウェブサイトへのアクセス，入場券，決済サービス，アカウントの個人情報の伝達などの様々な用途での利用が広まっている．

QR コードは高い認識率を誇るものの，単に黒と白のモジュールが並べられた画像データであるため，それに格納された内容を人は直接データとして解釈できない．つまり，利用者は QR コードの全体もしくは一部が書き換えられていたとしても気づくことができない．また，QR コードに格納されたデータは正しいと，むやみに信用してしまう利用者も多い．これを利用して，悪意のあるものが偽装した QR コードを作成し，それを読み取った利用者の不用意な操作により悪意のあるサイトに導かれることが問題となっている．特に，QR コードを決済に用いることが一般化した中国では，店舗側に表示された QR コードを第三者が張り替えることで，不正送金させる事件が起こっている [3]．そのほかにも QR コードの貼り換えや一部書き換えによる攻撃シナリオが想定される [4][5]．悪意のある QR コードの攻撃シナリオは自動化されたプロセス対象とするものと人を対象とするものに分類される．自動化プロセスを対象とするもには，SQL インジェクション攻撃やブラウザベースのエクスプロイト攻撃やクロスサイトスクリプティング攻撃などがある．人を対象とするものにはフィッシングや詐欺，そしてマルウェア感染などの危険がある．一方で，QR コードの貼り換えや書き換えにより異なるデータを格納した偽装 QR コードを読み取る場合，必ず悪意のあるサイトや情報が出力される．そのため，QR コードの設置者は事前にあるいは定期的に，QR コードを読み取って情報を確認すれば，偽装 QR コードの発見は容易であり早い段階での対策が可能である．決済サービスや認証サービスなどでの利用が進む QR コードの安全性を議論する上で，まだ明らかになっていない脆弱性や攻撃方法を発見することは非常に重要である．

1）神戸大学大学院工学研究科, Graduate School of Engineering, Kobe University

図 1　提案する方法で作成した偽装 QR コードの一例

本研究では発見が容易でない偽装 QR コードの作成方法を明らかにする．悪意のあるデータのみが格納された偽装 QR コードからは必ず悪意のあるデータが出力されるため，数回確認すれば偽装 QR コードの発見は容易である．我々はすでに，大きな確率 p で正規の URL を出力し，小さい確率 $1-p$ で悪意のある URL を出力する偽装 QR コードの作成方法を文献 [8] で与えている．この偽装 QR コードは悪意のある URL が小さい確率でしか出力されないため，QR コード設置者が数回確認したとしても発見できない．また，悪意のある URL が出力され被害にあった被害者が，もう一度偽装 QR コードを読み取ったとしても正規の URL が表示されるため，再現性がなく，偽装 QR コードが被害の原因であると気づくことができない．悪意のあるシステム会社や作成サイトにこの方法で偽装 QR コードを生成されてしまうと，偽装 QR コードの発見，対策が遅れ，被害の拡大が考えられる．

文献 [8] では，正規の URL とその URL の一文字を変化させた URL を出力する QR コードを作成している．本稿では，文献 [8] の手法を一般化して，任意の URL と別の任意の URL を出力する QR コードを作成する方法を与える．すなわち，まったく同一の QR コードで，ある確率 p でデータ A を出力し，確率 $1-p$ で異なるデータ B を出力する QR コードの作成方法を開発する．QR コードには，QR コードの破損や汚れ，あるいは撮影時の影などのノイズに耐性を持たせるために誤り訂正符号が用いられている．誤り訂正符号はそのパラメータによって訂正可能な誤りの数（誤り訂正能力）は決まっており，QR コードの読み取り時に生じた誤りが誤り訂正能力以下であれば正しい情報が出力される．誤り訂正能力を超える誤りが生じた場合には，読み取り不可能（誤り検出）または別の情報が出力されること（復号誤り）になる．文献 [8] での実験結果から，ノイズのない正規の QR コードの読み取りの際には復号誤りとなることはほぼ無いことがわかっている．本稿では，QR コードの一部を誤り訂正能力の限界まで書き換えた上で，更にノイズを一つあるいは複数のモジュールに追加し，意図

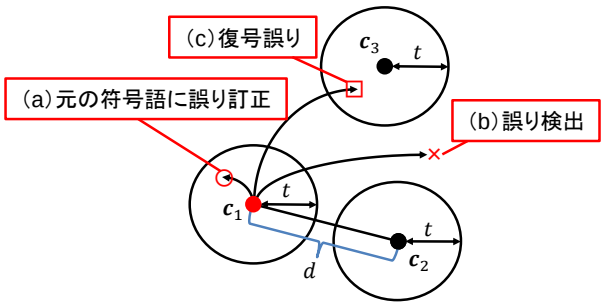

図 2　誤りの数と復号結果の関係

的に低い確率で復号誤りを起こす QR コードを構成する．構成した QR コードは復号誤りが起きない場合と起きる場合で別の情報を出力する．図 1 はその一例であり，ある一定の確率で悪性サイトの URL を出力する．なお，出力する悪性サイトの URL は本研究で実験的に作成した無害の Web ページである．評価実験では，この方法で作成した QR コードが任意の確率で悪性サイトの URL が出力することを示し，QR コードの新たな危険性を明示するとともに注意を喚起する．

2　誤り訂正符号と QR コード

2.1　RS 符号

Reed-Solomon(RS) 符号は誤り訂正符号の一つで，QR コードや CD，デジタル放送，衛星通信などに利用されている [6]．誤り訂正符号とは，送信する情報に冗長ビットを付加することで，通信中に発生した誤りを訂正し，正しいデータを復元する技術である．また，通信だけでなく保存された情報の信頼性が重要となる磁気記録媒体や CD，DVD などの記録媒体にも利用されている．情報に冗長ビットを付加することを符号化といい，冗長ビットが付加された系列を符号語と呼ぶ．RS 符号は符号語をビットの集まり（シンボル）で表し，シンボル単位で誤り訂正を行うのため連続して起こるビット誤り（バースト誤り）に強いという特徴を持つ．QR コードを読み取る際に生じる誤りは，QR コードの一部の汚れや損傷など連続して起こる誤りが多く，RS 符号と相性がよい．

符号長 n，情報点数 k の (n,k)RS 符号は，k シンボルで表された情報系列を $n(>k)$ シンボルの符号語系列に符号化する．(n,k)RS 符号において，ユークリッド復号法 [6] を用いて訂正可能な誤りの数は設計距離 $d=n-k+1$ により保証されており，訂正可能な誤りシンボルの数（誤り訂正能力）t は，

$$t=\left\lfloor \frac{d-1}{2} \right\rfloor = \left\lfloor \frac{n-k}{2} \right\rfloor \tag{1}$$

で与えられている．すなわち，誤り訂正能力 t の符号を用いると，受信した系列（受信語）に生じた誤りを t シンボルまで訂正可能である (図 2(a))．t シンボルを超える誤りが生じた場合は，誤り検出 (図 2(b)) もしくは復号誤り (図 2(c)) となる．復号誤りは，受信語と元の符号語以外の符号語の距離が t 以下となる場合に生じ，復号時に誤りを検出することなく，別の符号語に誤り訂正されてしまう．RS 符号は線形符号の一つであり，任意の二つの符号語 $\boldsymbol{c}_1$，$\boldsymbol{c}_2$ の線形結合 $\boldsymbol{c}_1+\boldsymbol{c}_2$ も RS 符号の符号語となる．また，RS 符号の検査行列 H と符号語 $\boldsymbol{c}$ に対して，

$$\boldsymbol{c}\cdot \mathrm{H}^{\mathrm{T}}=\boldsymbol{0} \tag{2}$$

が成立する．ここで，H は $(n-k)\times n$ 行列であり，$\boldsymbol{c}$ は n 次元ベクトルである．H^{T} は H の転置行列を表す．

QR コードでは，8 ビットを一つのシンボルとして扱う GF(2^8) 上の RS 符号が用いられる [1]．n や k などのパラメータは，原則として QR コードに格納したい情報の大きさを基に決定されるが，作成者が任意のパラメータを選択することも可能である．高速な読み取りを実現するために，QR コードから読み取った系列の復号には計算量の小さいユークリッド復号法が用いられている．破損や汚れの無い QR コードを読み取ったときに誤りが生じることはほぼなく，一般的な利用の範囲で復号誤りが生じることはほぼないと考えられる．

2.2　QR コード

QR コードは格納情報を高速かつ正確に読み取りが可能な二次元コードで，1994 年に株式会社デンソーの開発部門（現在は分離し株式会社デンソーウェーブ）によって開発された [2]．QR コードの誤り訂正レベルは小さい方から L，M，Q，H の順で 4 段階に設定でき，最も高いレベル H では QR コードの約 30 %が損傷していても格納情報を正確に読み出せる．誤り訂正のレベルを上げれば誤りに対する耐性が向上するが，誤り訂正のための冗長なシンボルが多く必要であり，格納するデータの容量が小さくなる．一般的に使用されているモデル 2 の QR コードには，1 型（21 × 21 モジュール）から 40 型（177 × 177 モジュール）までの型番が用意されており，バイナリデータは 2953 バイトまで格納することができる [1]．本稿では，型番 2，誤り訂正レベル M の QR コードを 2-M 型 QR コードと表記する．

二次元コードである QR コードは正方形の碁盤の目（モジュール）に白黒を配置する方法で作成される．QR コードは位置検出パターン，タイミングパターンなどの機能パターンと，情報ブロック，誤り訂正ブロック，形式情報などの符号化領域で構成される [7]．QR コードは以下の流れで生成される．

[格納する情報 (符号語) の生成と QR コードへの格納]

Step 1:　QR コードに格納する文字列を指定の文字コードで二進数に変換する．図 3 では，URL を 8 ビットコードに変換している．

Step 2:　二進数化したデータの先頭にモード指示子・文字数指示子を，末尾に終端パターンを付加し，情報コードとする．図 3 では，8 ビットコードを表すモード指示子「0100」，22 文字を表す文字数指示子「00010110」を先頭に付加し，終端パターンとして「0000」を末尾に付加している．

Step 3:　情報コードのシンボル数が格納する QR コードの情報ブロックの容量に満たなければ，足りない分だけ埋め草コードを付加する．図 3 では，2-M 型 QR コードの情報ブロックの容量が 28 シンボルであり，情報コード語が 24 シンボルのため，情報コードの末尾に埋め草コードを 4 シンボル付加している．

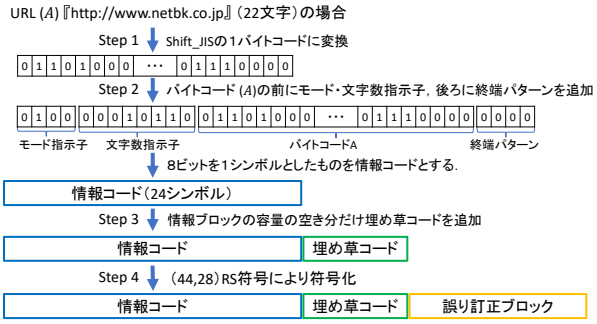

図 3　格納する情報（符号語）の生成例

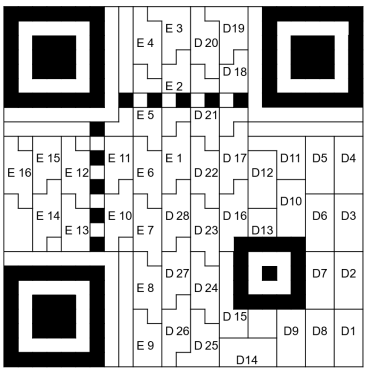

図 4　2-M 型 QR コードの配置方法

Step 4:　情報コードと埋め草コードを合わせた情報ブロックを RS 符号で符号化し，誤り訂正ブロックを情報ブロックの末尾に付加する．図 3 では，(44,28)RS 符号で符号化している．

Step 5:　符号語を図 4 に従って QR コードの右下から D1,D2 と順に配置する．

3　異なる二つの情報を出力する QR コード

本章では，誤り訂正符号の復号誤りの性質を用いて，まったく同一の QR コードで，ある確率 p でデータ A を出力し，確率 $1-p$ で異なるデータ B を出力する QR コードの作成方法を提案する．

3.1　復号誤りの制御

(n,k)RS 符号を用いたとき，任意の二つの符号語間の距離の最小値は $d=n-k+1$ で保証されており，ユークリッド復号法によって訂正可能な誤りの数は $t=\lfloor (d-1)/2 \rfloor$ である．このとき，データ A の符号語 $\boldsymbol{c}_A$ に誤り $\boldsymbol{e}$ が生じて，データ B の符号語 $\boldsymbol{c}_B$ に復号される条件は $\boldsymbol{c}_A+\boldsymbol{e}$ と $\boldsymbol{c}_B$ の距離が誤り訂正能力 t 以下になることである．$\boldsymbol{c}_A$ と $\boldsymbol{c}_B$ の距離を $d(A,B)$ とすると，この条件を満たすためには最低でも $d(A,B)-t$ シンボルが符号語 $\boldsymbol{c}_B$ のシンボルと一致する誤りが生じる必要がある．一般の撮影環境下で，この条件を満たす誤りが発生する確率は，$d(A,B)=d$ であったとしても限りなく低い．任意の二つの符号語間の距離の最小値は d で保証されているが，実際の符号語の距離はそれよりも大きくなる．そのため，ノイズのない QR コードを読み取って復号誤りを起こすことはほとんどない．

そこで，符号語 $\boldsymbol{c}_A$ のシンボルをあらかじめ t シンボルと一致するように変更して QR コードに格納すること

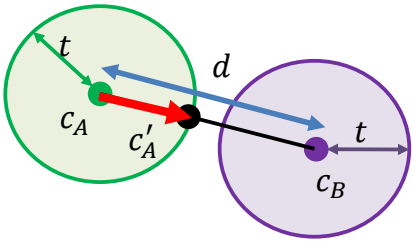

図 5　符号間距離

を考える．その系列を $\boldsymbol{c}'_A$ とすると符号語 $\boldsymbol{c}_B$ との距離は $d(A,B)-t$ であり，系列 $\boldsymbol{c}'_A$ の $(d(A,B)-t)-t$ シンボルが $\boldsymbol{c}_B$ と一致する誤りが生じることで，復号誤りが発生する．特に，$d(A,B)=d$ で奇数のとき，$d-2t=1$ であり，$\boldsymbol{c}'_A$ のある 1 シンボルが符号語 $\boldsymbol{c}_B$ のシンボルと一致する誤りが生じれば復号誤りが起こる．QR コードの 1 シンボルは 8 ビットで構成されており，あるシンボルが $\boldsymbol{c}_B$ のあるシンボルと一致する確率はビット反転確率 q としたとき，最大で q^8 であり，$q=1/2$ だとしても，0.4% 程度である．一般の撮影環境下でビットが反転する，すなわち黒モジュールを白と読み取る，あるいは白モジュールを黒と読み取ることはほぼなく，指定のビットが反転する確率は更に小さい．そこで，文献 [8] で反転させたいモジュールの中心に黒点（ドット）を付与することで，ビット反転する確率を制御できることを示している．これを利用すれば，任意のモジュールでの誤りの生じやすさを制御できるため，任意の二つのデータの符号語間の距離が必ずしも最小距離 d となる必要はない．

誤り制御のために付与するノイズはドットに限らず様々な方法が考えられる．一般に普及している QR コードリーダで，モジュールの白・黒の判定に利用される主な要素は以下の二つである．

要素 1　対象のモジュールの中心の輝度値
要素 2　周囲の複数モジュールの輝度値を参照して決定する白・黒を判定する閾値

この二つの要素を考えると，文献 [8] のように，反転させたいモジュールにドットを配置する，あるいは対象モジュールの輝度値を白と黒の中間（灰色）にすることで，白・黒の判定を不安定にできる．例えば，モジュールの中心にドットを置いた場合は要素 1 が影響し，撮影時のブレや傾きなどでデコーダ内部でのモジュールの中心がずれたときに，白・黒の判定が揺れる．対象モジュールの輝度値を灰色にする場合は要素 2 が影響し，撮影時の明るさや影などで白・黒の判定が揺れる．さらに，配置するドットの大きさや位置，またはモジュール全体の輝度値を調節することで，対象モジュールを白と読み取る確率や黒と読み取る確率を制御することが可能である．

以上のことをまとめると，データ A とデータ B の二つの情報を出力する QR コードを作成する手順は以下の通りである．

[異なる二つの情報を出力する QR コードの作成方法]

Step 1:　データ A とデータ B の符号語 $\boldsymbol{c}_A$ と $\boldsymbol{c}_B$ を生成する．$\boldsymbol{c}_A$ と $\boldsymbol{c}_B$ の距離を $d(A,B)$ とする．

Step 2: 符号語 $\boldsymbol{c}_A$，$\boldsymbol{c}_B$ の異なる $d(A,B)$ シンボルの中から，任意の t シンボルを選択し，符号語 $\boldsymbol{c}_A$ のシンボルを符号語 $\boldsymbol{c}_B$ に置き換えて，系列 $\boldsymbol{c}'_A$ を作る．

Step 3: 系列 $\boldsymbol{c}'_A$ を QR コードの生成方法に従って，白黒のモジュールを配置する．

Step 4: 系列 $\boldsymbol{c}'_A$ と符号語 $\boldsymbol{c}_B$ の異なる $d(A,B)-t$ シンボルの中から $d(A,B)-2t$ シンボルを選択し，ノイズを付与する．

このとき，Step 1 で生成する符号語間の距離が大きすぎると復号時に誤り検出となる確率が高くなり，QR コードの読み取り速度が正規の QR コードよりも遅くなることに注意する必要がある．また，Step 4 で付与するノイズは，対象モジュールにドットを置くものや対象モジュールを灰色にするものなどであり，ノイズを付加するシンボルの数などに応じて調節する．

3.2 埋め草コードを利用した符号語の生成

本節では，QR コードの埋め草コードを利用して，データ A とデータ B の符号語間の距離ができるだけ小さくなるように符号化する方法を与える．

QR コードはバージョンや誤り訂正レベルに従って，利用する RS 符号のパラメータが決まっている．そのため，情報コードのシンボル数が情報ブロックのシンボル数に満たないとき，埋め草コードが挿入される．埋め草コードは無為なデータであり，任意のデータへの入れ替えが可能である．つまり，二つのデータの符号語間の距離が小さくなるように埋め草コードを埋め込むことができる．

問題を簡単にするために，RS 符号の線形性を利用する [6]. RS 符号は線形符号であり，$\boldsymbol{c}_A+\boldsymbol{c}_B$ も RS 符号の符号語である．このとき，二つのデータ A，データ B の符号語は，二つのデータの排他的論理和をとったデータ C の符号語と等しくなるため，$\boldsymbol{c}_A+\boldsymbol{c}_B=\boldsymbol{c}_C$ とおく．そして，二つの符号語 $\boldsymbol{c}_A$ と $\boldsymbol{c}_B$ の距離は，$\boldsymbol{c}_C$ の非零のシンボル数と等しくなる．ここで，演算子 + はガロア体 GF(2^8) のシンボル同士の和を表す．すなわち，二つの符号語 $\boldsymbol{c}_A$ と $\boldsymbol{c}_B$ の距離を小さくすることは，$\boldsymbol{c}_C$ の非零のシンボル数を小さくすることと等しい．したがって，$\boldsymbol{c}_C$ の非零のシンボル数をできるだけ小さくする符号化方法を与える．

まず，あるデータ C が与えられたとき，検査行列を用いて符号語を生成する方法を説明する．データ C から生成した情報コードを $\boldsymbol{i}$ とし，埋め草コードを $\boldsymbol{u}$，誤り訂正ブロックを $\boldsymbol{p}$ とする．(n,k)RS 符号の検査行列を H とすると，符号語 $(\boldsymbol{i}|\boldsymbol{u}|\boldsymbol{p})$ について，次の式が成立する．ここで，| はシンボルの連結を表す．

$$(\boldsymbol{i}|\boldsymbol{u}|\boldsymbol{p})\cdot \mathrm{H}^{\mathrm{T}}=\boldsymbol{0} \tag{3}$$

ここで，$\boldsymbol{i}$ が既知であることから，式 (3) を変形すると，

$$(\boldsymbol{0}|\boldsymbol{u}|\boldsymbol{p})\cdot \mathrm{H}^{\mathrm{T}}=(\boldsymbol{i}|\boldsymbol{0}|\boldsymbol{0})\cdot \mathrm{H}^{\mathrm{T}} \tag{4}$$

と書ける．$\boldsymbol{b}=(\boldsymbol{i}|\boldsymbol{0}|\boldsymbol{0})\cdot \mathrm{H}^{\mathrm{T}}$ と置くと，

$$(\boldsymbol{0}|\boldsymbol{u}|\boldsymbol{p})\cdot \mathrm{H}^{\mathrm{T}}=\boldsymbol{b} \tag{5}$$

となり，$(\boldsymbol{u}|\boldsymbol{p})$ の各シンボルを変数とする連立一次方程式とみなせる．したがって，あるデータ C が与えられたときに符号語を生成することは，式 (5) を解いて $(\boldsymbol{0}|\boldsymbol{u}|\boldsymbol{p})$ を求めることと等しい．

次に，この方法で非零のシンボル数が d_0 となる符号語を生成することを考える．情報コードのサイズを $k'(<k)$，非零のシンボル数を d_1 とするとき，符号語の非零シンボル数が d_0 となるためには，$(\boldsymbol{0}|\boldsymbol{u}|\boldsymbol{p})$ の非零のシンボル数が d_0-d_1 となる必要がある．ここで，$(\boldsymbol{u}|\boldsymbol{p})$ のうち，d_0-d_1 個の非零シンボルの系列を $\boldsymbol{x}$ とする．さらに H′ を H の列ベクトルのうち，$(0|\boldsymbol{u}|\boldsymbol{p})$ の非零シンボルと同じ位置の列ベクトルのみで構成される $(n-k)\times(d_0-d_1)$ 行列とする．このとき，式 (5) は，

$$\boldsymbol{x}\cdot \mathrm{H}'^{\mathrm{T}}=\boldsymbol{b} \tag{6}$$

と書き直すことができる．これは d_0-d_1 個の変数を持つ連立方程式と考える事ができ，解を持つ条件は

$$\mathrm{rank}\ \mathrm{H}'=\mathrm{rank}\ [\mathrm{H}'|\boldsymbol{b}^{\mathrm{T}}] \tag{7}$$

が成り立つことである．そして，解を持つならば，非零のシンボル数が d_0-d_1 である $(0|\boldsymbol{u}|\boldsymbol{p})$ を生成できる．つまり，非零シンボル数が d_0 の符号語を生成するためには，式 (6) が解を持つように $(0|\boldsymbol{u}|\boldsymbol{p})$ の非零シンボルを決めれば良い．

以上のことを利用して，データ C が与えられたとき，できるだけ小さいの符号語を以下の手順で生成する．

[埋め草コードを利用した符号語の生成方法]

Step 1: データ C の情報コード $\boldsymbol{i}$ を作り，そのサイズ $k'(<k)$ と非零のシンボル数 d_1 を求める．

Step 2: 最小距離 $d>d_1$ の (n,k)RS 符号を用いる．目標とする非零シンボル数を $d_0=d$ とする．

Step 3: $(\boldsymbol{0}|\boldsymbol{u}|\boldsymbol{p})$ の非零のシンボルを d_0-d_1 個決めて，式 (6) が解を持つかどうかを式 (7) でチェックする．解を持つ場合，次の Step に移る．解を持たない場合，非零とするシンボルを選択し直す．全ての組み合わせで解を持たない場合，目標の非零シンボル数 d_0 を 1 増やす．

Step 4: 式 (6) を解き，符号語 $\boldsymbol{c}_C=(\boldsymbol{i}|\boldsymbol{u}|\boldsymbol{p})$ を求める．

Step 3 の全ての組み合わせの数は $n-k'$ シンボルの中から d_0-d_1 シンボルを選ぶ組み合わせの数となる．

データ A とデータ B が与えられたとき，それぞれの情報コード $\boldsymbol{i}_A,\boldsymbol{i}_B$ の線形結合 $\boldsymbol{i}_A+\boldsymbol{i}_B$ を情報コード $\boldsymbol{i}$ として上記の方法で非零のシンボル数 d_0 の符号語が生成できたとする．$\boldsymbol{c}_A+\boldsymbol{c}_B=\boldsymbol{c}_C$ が成立するため，$\boldsymbol{c}_A$ を正規の方法で生成した後，$\boldsymbol{c}_B$ を $\boldsymbol{c}_B=\boldsymbol{c}_C+\boldsymbol{c}_A$ で求めることで，二つのデータの符号語間の距離が d_0 となるように符号化できる．

特殊ケースとして，$\boldsymbol{i}$ の非零のシンボル数が $d_1=1$ のとき，すなわち，データ A とデータ B の情報コードが 1 シンボルのみ異なるときを考える．このとき，埋め草コード $\boldsymbol{u}=\boldsymbol{0}$ として符号化すると，RS 符号の最小距離が $d=n-k+1$ であることから，誤り訂正ブロックの $(n-k)$ 個のシンボルが全て非零となる．そして，$\boldsymbol{c}_c$ の非零のシンボル数は明らかに d である．この場合，データ A とデータ B を正規の方法で符号化すれば，その符号語間の距離は最小距離 d となる．

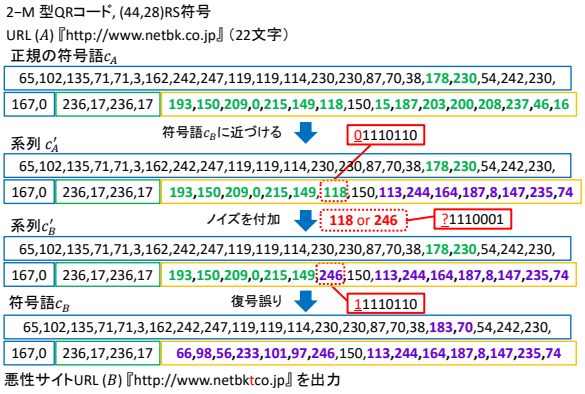

図 6　偽装から悪性サイトへ誘導までの流れ

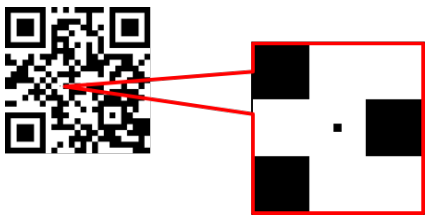

図 7　ノイズ（ドット）の付与

4　偽装 QR コードの作成と評価実験

本章では，提案した二つの情報を出力する QR コードを悪用して，小さい確率 $1-p$ で悪性サイトに誘導する QR コードを作成できることを示し，その危険性を明らかにする．

4.1　偽装 QR コードの作成

本稿では，住信ネット銀行 [9] のトップページの URL(http://www.netbk.co.jp) を対象とする．提案した QR コードの生成方法で，小さい確率 $1-p$ で 1 文字異なる URL(http://www.netbktco.jp) を出力する偽装 QR コードを作成することが可能なことを説明し，提案した QR コードの危険性を明らかにする．これらの二つの URL 格納する 2-M 型 QR コードには (44,28)RS 符号が利用され，その最小距離は $d = 17$ であり，誤り訂正能力は $t = 8$ である．正規の URL と悪性サイトの URL の情報コードの違いは 1 シンボルのみであり，3.2 節の特殊ケースに当たる．したがって，二つの符号語間の距離が最小距離 d であり，図 5 の関係にある符号語 $\boldsymbol{c}_A, \boldsymbol{c}_B$ を生成できる．以下の流れで偽装 QR コードを作成する．図 6 に符号語のどの部分を変更しているか示している．図中では各シンボル（8 ビット）を 10 進数で表記している．

[偽装 QR コードの作成例]

Step 1:　正規サイト (http://www.netbk.co.jp) と悪性サイト (http://www.netbktco.jp) を正規の方法で符号化し，$\boldsymbol{c}_A$ と $\boldsymbol{c}_B$ を生成する．この二つのデータの情報コードは 1 シンボルのみ異なり，$\boldsymbol{c}_A$ と $\boldsymbol{c}_B$ の距離は $d(A,B) = d = 17$ となる．

Step 2:　符号語 $\boldsymbol{c}_A$ と $\boldsymbol{c}_B$ 間で異なる 17 シンボル（図 6 中の太字）の中から $t = 8$ シンボルを選択し，符号語 $\boldsymbol{c}_A$ のシンボルを符号語 $\boldsymbol{c}_B$ の同じ位置のシンボルに置き換えて，系列 $\boldsymbol{c}'_A$ を作成する．

Step 3:　系列 $\boldsymbol{c}'_A$ を図 4 の配置に従って配置する．

表 1　実験環境

撮影場所	白色蛍光灯下
プリンタ	Canon Pixus MP610
印刷用紙	普通紙（白色率 68 %）
QR コードの大きさ	$2.65 \times 2.65 cm^2$
1 モジュールのサイズ	7×7 ピクセル

表 2　撮影機器の性能

撮影機器（デコーダ）	iPhone 7（QR コードリーダー for iPhone）
解像度	4,032×3,024 ピクセル
画素数	1200 万画素

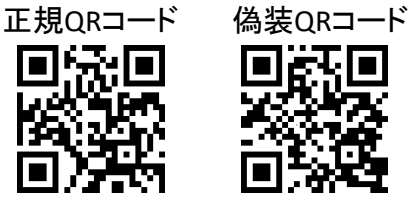

図 8　正規の QR コードと偽装 QR コードの比較

図 9　ノイズを付与した偽装 QR コード（ドットの輝度値 160）

Step 4:　系列 $\boldsymbol{c}'_A$ と符号語 $\boldsymbol{c}_B$ の異なる $d - t = 9$ シンボルの中から 1 シンボル（図 6 中の点線の枠）を選択し，ノイズを付与する．

Step 4 で選択したモジュールに対して，図 7 のようにモジュールにドットを配置を付与して偽装 QR コードを作成した．このモジュールを白と読み取った場合には正規サイトの URL が出力され，黒と読み取った場合には悪性サイトの URL が出力される．図 8 は正規の QR コードと偽装 QR コードを並べて比較したものである．利用者は QR コードの内容を直接理解できないため，QR コードが偽装されていることに気づきにくい．また，ドットに気づいたとしても，それが自然にできた汚れか意図的につけられたものかを判断することはできない．

4.2　復号誤りの制御の評価

本節では，前節で生成した偽装 QR コードの読み取り実験を通して，復号誤りが生じる確率をどの程度制御できるかを評価する．

4.2.1　評価方法と評価結果

文献 [8] では，モジュールの中心に付与したドットの輝度値を 0（黒）から 255（白）まで変化させて，復号誤りが起こる確率を評価している．図 9 は，ドットの輝度値を 160 とした偽装 QR コードであり，復号誤りが約 1/100 で生じる．また，ドットの輝度値が 255，すなわちドットを付与しない場合は復号誤りは生じなかった．これは実環境において，ノイズの無い QR コードの 1 モ

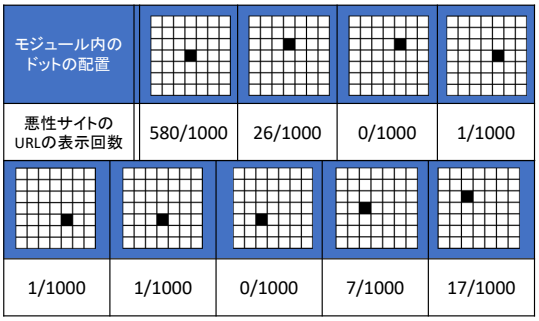

図 10　ドットの配置を変化させた場合の読み取り結果

ジュールの白黒を反転して読み取ることがほぼ無いことを示している．

本稿ではドットをモジュールのどこに配置するかによって，復号誤りがどの程度制御できるかを読み取り実験を通して確認する．ドットの配置を変えた偽装 QR コードを 1000 回ずつ読み取り，復号誤りが生じた回数をカウントする．付加するドットのサイズは 1×1 ピクセルに固定する．読み取り実験のデコーダは，一般に広くインストールされている『QR コードリーダー for iPhone(iPhone7)』を使用する．実環境を想定して，白紙に印刷した偽装 QR コードを白色蛍光灯下の室内で様々な角度・距離から撮影する．実験環境は表 1，撮影機器の性能は表 2 の通りである．

読み取り結果を図 10 に示す．ドットを中心に配置した場合，約 3/5 の確率で悪性サイトの URL が表示されている．一方，ドットを中心からずらした場合，1/1000 などの低確率で悪性サイトの URL が表示されている．また，ドットを中心から右上などの斜め方向にずらす場合は，1 度も表示されないことがある．この結果は，QR コードリーダがモジュールの白，黒を判定する要素の一つである対象のモジュールの中心の輝度値が関係している．読み取り時のブレなどで走査線が中心からずれた場合に，判定が変わっていると思われる．斜め方向のドットは 0 回の場合もあったが，試行回数を 1 万回，十万回と増やすことで読み取れる可能性も大いに考えられる．この実験により中心に配置したドットの輝度値だけでなく，ドットの配置位置を変えることでも確率を制御できることを確認できた．

4.2.2　偽装 QR コードの危険性

提案した偽装 QR コード上のノイズ（ドット）は非常に小さく見えづらいため，正規の QR コードと比較してもそれが偽装されているものだと気づくことは容易ではない．また，今回のように一文字だけ違う URL を表示させる場合や短縮 URL を用いたサイトの場合，出力された URL を一瞬だけ見ただけでは改ざんされていることに気づきにくい．さらに，QR コードの撮影時に自動的にサイトを表示させる場合，気づかない間に悪性サイトに誘導され，非常に危険である．そして，この偽装 QR コードは小さな確率で悪意のあるサイトに誘導されることから，詐欺やマルウェア感染などの被害にあったとしても再現が難しく，発見を遅らせることになり被害の拡大となることが予想される．

さらに，モジュール全体の輝度値を変化させて，作成した偽装 QR コードが図 11 である．この QR コードは正規のサイトとして，(http://srv.prof-morii.net/~lab) が格納されており，悪性サイトとして (http://srv.prof-morii.net/~lob) が格納されている．この偽装 QR コードも同様に，小さい確率で悪性サイトの URL を表示させることができる．単一のモジュールだけの輝度値が変わっているとわかりやすいが，複数モジュールの色を変えたり，イラストに重ねて印刷したりすれば目立たなくなる．

図 11　モジュールの輝度値を変化させて作成した偽装 QR コード

5　まとめ

本稿では，QR コードの破損や汚れに耐性を持たせるために利用されている誤り訂正符号の性質である復号誤りを用いて，異なる二つのデータを出力する QR コードの生成方法を明らかにした．そして，その方法を悪用することでほとんどの場合は正規のサイトに誘導されるが小さい確率で悪性サイトに誘導される偽装 QR コードを作成できることを示した．QR コードは人が直接，その意味を解釈できないことから，今回の提案による偽装された QR コードの発見は必ずしも容易ではない．特に悪意のある QR コード作成者によって，このような偽装された QR コードが作成，配布された場合，大きな被害が想定される．今後は QR コードの信頼性について問題とするとともに，利用者は QR コードをむやみに信頼するのではなく，必ず認識した URL を確認してから Web サイトへ遷移するべきである．

参考文献

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

[2] DENSO WAVE INCORPORATED, http://www.qrcode.com/index.html, visited on June 26, 2018.

[3] 牧野武文，”シールを貼るだけのお手軽詐欺 アリペイの偽 QR コードで 1 万元を盗む”, https://the01.jp/p0005594/, sprout，August 21 2017, visited on June 26, 2018.

[4] P. Kieseberg, M. Leithner, M. Mulazzani, L. Munroe, S. Schrittwieser, M. Sinha, E. Weippl, “QR Code Security”, Proc. in TwUC’ 10, Paris, France., Nov. 8–10, 2010.

[5] K. Krombholz, P. Frühwirt, P. Kieseberg, I. Kapsalis, M. Huber, E. Weippl, “QR Code Security: A Survey of Attacks and Challenges for Usable Security”, Lecture Notes in Computer Science, 8533(2014), pp.79–90.

[6] 今井秀樹, ”符号理論”, 電子情報通信学会, pp.155-173, 1990.

[7] 池田和興, ”例題が語る符号理論　BCH 符号・RS 符号・QR コード”, 共立出版, 2007.

[8] 大熊浩也, 瀧田　愼, 森井昌克, “悪性サイトに誘導する QR コードの存在とそれを利用した偽造攻撃”, 信学技報, vol. 118, no. 109, ICSS2018-6, pp. 33-38, 2018 年 6 月.

[9] 住信 SBI ネット銀行 トップページ，http:www.netbk.co.jp, May 29 2018.