L-006

オフラインサイト認証可能なQRコード

Two-dimensional code with site authentication off-line

柏井　祐樹 †　　渡辺　優平 ‡　　森井 昌克 ‡
Yuki Kashii　Yuhei Watanabe　Masakatu Morii

1　はじめに

QR コード [1] は代表的な 2 次元コードの一種であり，多くの WEB サイトや広告紙面上に掲載されている．また，市販されている多くの携帯電話は QR コードデコーダを搭載しており，誰もが気軽に利用することができる．しかし，QR コードは格納データの視認性に乏しく，偽造された QR コードの判断が困難である．さらにユーザが読み取ったデータをきちんと確認することなく WEB サイトにアクセスすることが多い．そのため，QR コードを介して不正な WEB サイトに誘導される危険性が存在する．この対策として認証サーバを用いたオンラインでの認証方式 [2] が提案されている．しかし，認証サーバを用いた方式ではデコードの際に時間を要することや通信路に対する負荷が増大するという問題が生じる．

本稿では，不正 QR コード対策を施した QR コードの生成方法およびサーバと通信しない認証方式を提案する．提案方式では QR コードの下部に格納データと対応する URL を記載する．この URL は OCR を用いて読み取りを行う．OCR の読み取り結果に生じる誤りを訂正するために，記載した URL から符号語を生成する．さらに，この符号語に対して認証局により電子署名を付加し，記載した URL と符号語に対しての信頼性を確保する．署名は埋め草コード語の格納領域に埋め込む．そして，QR コードを読み取る際に OCR の読み取り結果に誤りが生じているか否かを確認する．発生した誤りの割合から QR コードの格納データである URL の認証を行う．以上の手順で作成した QR コードは従来のデコーダで規定の性能どおりの読み取りが可能である．さらに提案方式のデコーダではサーバと通信をすることなく高速認証可能となる．

2　QR コード

QR コードとは 1 次元に情報を格納する従来のバーコードと異なり，縦横の 2 方向に情報を保持することで，記憶できる情報量を飛躍的に増加させたマトリックス型 2 次元コードである．QR コードは破損や汚れがあるものを正確に読み取るために RS 符号 [3] を利用した誤り訂正の機能を有しており，格納されているデータを高速かつ正確に読み取れるという特徴を備えている．そのため，多くの広告紙面上に掲載されている．携帯電話に搭載されている QR コードデコーダを用いることにより，URL を直接入力することなく容易に WEB サイトにアクセスすることが可能となった．図 1 に神戸大学の URL から作成した QR コードを示す．図 1 から分かるように格納データの視認性に乏しく，偽造された QR コードの判断が困難である．

図 1　QR コード

2.1　QR コードの構成

以下で QR コードの各用語について解説する．

1. 型番
   QR コードのシンボルの大きさを表す．1 ～40 型まで存在し，型番が大きくなるにつれて格納可能なデータ量は増加するが，QR コード自体も大きくなってしまう．
2. モジュール
   QR コードを構成する最小の単位セル．1 モジュールが明暗の 2 値の状態を持ち，1 ビットの情報を表す．型番 N の QR コードの一辺に存在するモジュールの総数は $17 + 4 \times N$ の式により定まる．
3. モード
   ビット列として定義される格納データを文字に変換する方法．各々のモードごとに文字集合を表すビット列の長さが規定されている．数字モード，英数字モード，8 ビットバイトモードなど全部で 8 種類のモードが存在する．
4. 文字数指示子
   QR コードに格納されるデータ文字列の長さを定義するビット列．型番およびモードごとにビット列の長

† 神戸大学工学部，Faculty of Engineering, Kobe University
‡ 神戸大学大学院工学研究科，Graduate School of Engineering, Kobe University

さが定義されている．

5. コード語
   QR コードにおいてデータを 8 ビット毎に分割した 1 つあたりの塊を指す．格納データを表すコード語をデータコード語と呼ぶ．
6. 終端パターン
   データを表すビット列の終了に使用する 0000 のビットパターン．ビット列をコード語ごとに分割し，容量を完全に満たしている場合は終端パターンを省略する．コード語の残りの容量が 4 ビット未満の場合は短縮する．
7. 埋め草コード語
   データコード語の総数が最大データ量に満たない場合，格納データの末尾に終端パターンを付加した後で空のコード語を補填する目的で使用するコード語．埋め草コード語はデータを示さない固定のビットパターンで表現される．
8. RS 符号
   RS 符号とは QR コードの符号化の際に用いられる誤り訂正符号．連続して起こるビット誤りに強いので，CD,DVD などの様々なデジタル機器や通信分野における誤り訂正技術で応用されている．QR コードは 1 本の符号語または複数の符号語で構成される．複数の符号語を用いる場合は各符号語からデータが QR コード上にインターリーブ配置される．ある符号語において符号長 n，情報点数 k とすると最小距離 d_m は $n-k+1$ で与えられる
9. 誤り訂正能力
   QR コードが破損した場合に備えて，誤りの箇所を発見し訂正を行う機能．L，M，Q，H の 4 段階からなり，一番低い L で約 7 %，最も高い H で約 30 %の QR コードを構成する符号語に対して誤り訂正が可能である．
10. 型式
    型番と誤り訂正能力について記述したもの．例えば，10-M とは型番 10 の誤り訂正能力 M という意味である．

2.2 埋め草コード語

2.1 節より埋め草コード語は QR コードの格納データ量が最大データ量に満たない場合，空のコード語を補填する目的で使用する固定のビットパターンである．11101100 および 00010001 が交互に埋め込まれる．埋め草コード語の格納領域に埋め込まれたデータは QR コードの読み取りに影響しない．したがって，QR コードを構成する符号語が復号可能であれば，この領域にどんなデータを埋め込んでも問題ない．

2.3 認証サーバを用いた不正 QR コード対策

オンラインでの不正 QR コード対策として，認証サーバを用いた方式が提案されている．この方式における認証の手順を以下に示す．

step1　WEB サイトの認証登録
　認証 QR コードの作成を希望する者は URL を認証サーバに送信する．認証サーバは WEB サイトを検査し，正常な WEB サイトと判断した場合は URL とそれに対応する認証 ID を認証サーバに登録する．また検査の結果，不正 WEB サイトと判断した場合は認証 ID を発行せず，URL の登録も行わない．

step2　QR コードの作成
　step1 で発行した認証 ID を格納した QR コードを作成する．

step3　QR コードの読み取り
　WEB サイトへのアクセスを希望する者は携帯電話などのデコード機器を用いて QR コードを読み取り，認証 ID を取得する．

step4　WEB 参照
　QR コードから取得した認証 ID を認証サーバに送信する．認証サーバは送られてきた認証 ID から対応する URL を取得し，WEB サイトにアクセスしたい者に返送する．

以上から分かるように，認証サーバを用いた方式は QR コードを取得するだけでも時間と手間がかかり，QR コードの手軽に作成可能という利便性が失われている．さらに，QR コードを読み取るたびにサーバと通信する必要があるのでデコードに時間を要し，通信路に対する負荷が増大するという問題点が存在する．

3 電子署名

電子署名とはデジタル文章に現実の世界での印鑑の捺印やサインに相当する機能を果たすものである．電子署名を実現する技術として，公開鍵暗号方式を応用したデジタル署名方式が存在する．以下にデジタル署名の簡単なアルゴリズムを説明する．初めに送信者は検証鍵 pk と署名鍵 sk を生成する．pk は公開し，sk は送信者のみが保持する．検証鍵と署名鍵には図 2 の関係が成り立つものとする．図 2 から，片方の鍵で暗号化と復号の両方を行うことは不可能であることがわかる．また，pk から sk を求めることは困難とする．送信者はメッセージ M を sk を利用した署名生成機に入力して署名 m を取得する．そして M，m を受信者に送る．受信者は全体に公開されている pk と送信者から送られてきた M，m を署名検証機に入力する．検証機により正しい署名と判断された場合，

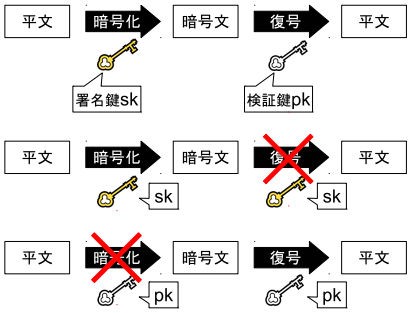

図 2　検証鍵と署名鍵の関係

送信者から送られてきたメッセージであると確認できる．以下にデジタル署名の 1 つである RSA 署名のアルゴリズムについて示す．

RSA 署名は公開鍵暗号の一種である RSA 暗号を用いたデジタル署名の方式であり，大きな数の素因数分解の困難性に基づいて安全性を保障している．まず，公開鍵および秘密鍵を生成するために 2 つの大きな素数 p と q $(p \neq q)$ を選択する．この時，安全性を最大にするために p と q を同じ長さにする．そして p と q の積を計算して，

$$l = pq \tag{1}$$

を得る．次に暗号化鍵 e を選ぶ．ただし，e と $(p-1)(q-1)$ は互いに素とする．そして，拡張ユークリッド互助法を用いて，

$$ed \equiv 1 \bmod (p-1)(q-1) \tag{2}$$

を満たす復号鍵 d を得る．ただし，l と d は互いに素とする．こうして得られた l，d，e から公開鍵 (e, l) と秘密鍵 (d, l) を生成する．実際に RSA 暗号を用いてメッセージの送受信を行う際には，まず受信者は先述した手順で生成した (e, l) を公開する．送信者は送りたいメッセージを m とすると，

$$c = m^e \bmod l \tag{3}$$

から暗号文 c を得る．受信者は c を受け取り，

$$m = c^d \bmod l \tag{4}$$

の計算を行い，元のメッセージ m を取得する．また，RSA 暗号は秘密鍵で暗号化して公開鍵で復号することも可能である．このことを利用して RSA 署名を実現する．RSA 署名の手順を以下に示す．

step1　鍵の生成

先述した手順でメッセージ M の送信者が公開鍵及び秘密鍵の生成を行う．公開鍵が検証鍵 pk，秘密鍵が署名鍵 sk となる．

step2　メッセージに署名を付加

RSA 暗号は秘密鍵でも暗号化可能なので，M に対して sk を用いて署名 m を生成する．

step3　受信者にデータの送信

生成した M，m を受信者に送信する．

step4　署名の検証

受信者は受け取った m を pk により復号し，M' を得る．そして，M と M' の比較を行い，一致した場合に正しいメッセージと判断する．

4　オフラインでの認証方式

提案方式では下部に格納データである URL を記載した QR コードを作成する．この URL は OCR を用いて読み取りを行う．OCR の読み取り結果に生じる誤りを訂正するために，記載した URL から符号語を生成する．さらに，この符号語に対して認証局により電子署名を付加し，記載した URL と符号語に対しての信頼性を確保する．署名は埋め草コード語の格納領域に埋め込む．そして，QR コードを読み取る際に OCR の読み取り結果に誤りが生じているか否かを確認する．発生した誤りの割合から QR コードの格納データである URL の認証を行う．さらに，提案方式により作成した QR コードは従来のデコーダでも規定の性能どおりの読み取りが可能である．以下にオフラインで認証可能な QR コードの作成手順と認証手順を示す．

4.1　作成手順

step1　URL の送信

認証 QR コードの作成を希望する者は WEB サイトの URL を，QR コードの埋め草コード語部に署名を格納する認証局へ送信する．

step2　符号語の生成に用いる URL の選択

QR コードの下部に記載する URL は誤って読み取る可能性を軽減させるために一部を省略する．具体的にはスキーム (http など)，ホストサーバ名 (www など)，URL の一番最後の「/」を省略する．前者 2 つは偽造が困難であり，「/」は URL の入力の際に省略しても問題がない．

step3　OCR の読み取り結果に対する符号語の生成

RS 符号を用いて符号語を生成する．生成する符号語は $(n, k, d_m) = (58, 52, 7)$ で固定する．URL の文字数と関係なく，この符号語により OCR での読み取り結果は 3 文字まで誤り訂正可能である．URL の文字数が増加すると読み取り結果に誤りが生じる可能性が高くなる．しかし，それに応じて誤り訂正可能な文字数を増加させると安全性が低下してしまう．なお，

[処理前]
kobe-u.ac.jp

[処理後]
kobe-u.ac.jp

図 3　ドットの処理

kobe-u.ac.jp

図 4　作成した QR コード

URL のデータ量がこの符号語の情報点数に満たない場合はデータとして意味を成さないコード語として，固定のビットパターン 11101100 および 00010001 を交互に埋め込む．

step4　署名の付加

認証局は受け取った URL を基に step3 で説明した手順で符号語を生成し，RSA 署名により符号語および URL に対して署名を付加する．署名に用いる鍵のサイズは 2048bit とする．公開鍵はデコーダに組み込む．

step5　URL の記載

QR コードの下部には step2 で説明した通り，一部を省略した URL を記載する．記載する URL の書体は Arial，サイズは 25 とする．さらに，OCR の読み取り精度上ドットがコンマやスペースと読み取られてしまうことが多い．そこで，ドットを本来の大きさより一回り大きく記載する．処理前と処理後の URL を図 3 に示す．

step6　QR コードの返送

認証局は完成した QR コードを URL の送信者に返送する．

神戸大学の URL を基にして作成した QR コードを図 4 に示す．次に，認証手順を記す．

4.2　認証手順

step1　QR コードデコーダおよび OCR で読み取り

デコーダで QR コードを読み取り，URL と埋め草コード語の格納領域から署名を取得する．OCR で QR コードの下部に記載された URL を読み取る．OCR で URL を読み取る際に，rgb 値を以下の式で輝度値 R に変換して 2 値化する．ただし，*red*，*green*，*blue* はそれぞれの色の濃さを 0～255(255 が最も濃い) で表しているとする．

$$R = (red \times 0.299 + green \times 0.587 + blue \times 0.114) \tag{5}$$

R が 90 未満なら黒，90 以上なら白として 2 値化を行う．

step2　署名の検証

デコーダに内蔵されている公開鍵を用いて署名を復号する．署名から URL を取得して QR コードの格納データと比較を行い，一致すれば署名が正しいと判断できる．

step3　OCR の読み取り結果の誤りを訂正

署名を復号して得られた符号語の情報点部に，OCR での読み取り結果を埋め込む．そして，OCR の読み取り結果の誤り訂正を行う．このとき，誤り訂正を行った箇所を取得する．

step4　認証

OCR での読み取り結果に誤りが生じなかった場合は正しい QR コードと判断する．誤りが生じた場合は誤り訂正を行った箇所をユーザに伝え，QR コードの読み取り結果が正しいかを確認するように警告する．誤り訂正能力以上の誤りが存在すると判断された場合は，ユーザに再度認証手続きを行うように促す．

5　オフラインでの認証 QR コードに対する評価と考察

提案方式により作成した QR コードを携帯電話で撮影し，オフラインでの認証が可能か否かの評価を行った．QR コードの型式は 11-L を用いた．OCR は google 提供のオープンソースである tesseract-ocr-3.01 を使用した．撮影は白色蛍光灯下において約 810 万画素の携帯電話で行った．携帯電話で撮影した QR コードを図 5 に示す．実験には多くのユーザが頻繁にアクセスする WEB サイトの URL を用いた．セキュリティ会社の Sophos が調査した結果から Apple，Google，Facebook では 1 文字異なる URL の 80 %以上が不正 WEB サイトに接続されてしまうことが確認されている [4]．このことから，正規のものと 1 文字異なる URL が格納されている不正 QR コー

図 5　撮影した QR コード

ドが高確率で存在すると考えられる．実験では 10 種類の QR コードを作成した．格納する URL は以下の通りである．10 種類のうち 5 種類の QR コードは正規のものを作成する．残りの 5 種類については QR コードの下部に記載する URL を正規のものと同様にし，格納する URL を正規のものと 1 文字異なるものとして不正 QR コードを作成する．実験に使用した QR コードの一例を図 6，図 7 に示す．

1. http://www.apple.com/
2. http://www.facebook.com/
3. http://www.google.co.jp/
4. http://www.twitter.com/
5. http://www.youtube.com/
6. http://www.apPle.com/
7. http://www.faceb0ok.com/
8. http://www.goog1e.co.jp/
9. http://www.tvvitter.com/
10. http://www.youfube.com/

表の番号は上記の URL の番号と対応している．それぞれの QR コードを 10 回撮影し，認証を行った．正しい QR コードの読み取り結果を表 1，不正 QR コードの読み取り結果を表 2 に示す．成功は OCR の読み取り結果に誤りが生じていないと判断された場合である．それに対して，警告は OCR の読み取り結果に 1 文字以上の誤りが存在すると判断された場合である．

それぞれの表から分かるように正しい QR コードに関しては少なくとも 1 回の成功が見られたのに対し，不正 QR コードでは 1 度も正しい QR コードと認証してしまうことはなかった．評価の結果から，提案方式により QR コードにおけるオフラインでの WEB サイト認証が実現できた．

図 6　正規の QR コード

図 7　不正 QR コード

表 1　正しい QR コードの読み取り結果

	1	2	3	4	5
成功	1 回	5 回	1 回	3 回	4 回
警告	9 回	5 回	9 回	7 回	6 回

表 2　不正 QR コードの読み取り結果

	6	7	8	9	10
成功	0 回	0 回	0 回	0 回	0 回
警告	10 回	10 回	10 回	10 回	10 回

6　まとめ

本稿ではサーバと通信することなく，認証を行う QR コードの生成方法および認証方式を提案した．提案方式では QR コードの下部に記載した URL を OCR で読み取った結果と QR コードのデータを表さない領域に埋め込んだ情報を比較することによってオフラインでの認証を行う．さらに，OCR での読み取り結果を保障するために誤り訂正を行う符号語を生成する．また，この符号語を自由に生成可能にすると悪用の危険性が存在する．こ

のため，認証局を通して符号語に対して符号語に署名を付加する．この署名を QR コードのデータを表さない領域に格納する．提案方式は正しい QR コードに対して少なくとも 1 回は認証が成功しているが，不正 QR コードを正規のものと認証することは 1 度もなかった．

今後の課題としては，RSA 署名方式を用いて実装を行ったので，署名のサイズが 2048bit と大きくなってしまった．その結果，QR コードの型式も大きなものを使う必要があった．電子署名方式を DSA など署名サイズが数百 bit のものを用いて実装を行い，より小さな型式でオフラインでの認証を可能にする．

参考文献

[1] 日本工業規格, “JIS X0510, 二次元コードシンボルー QR コードー基本仕様”

[2] 寺浦 信之, 櫻井 幸一, “二次元コードによる不正 WEB 誘導への対策”,SCIS 2012, 4E1-6, 2012 年

[3] 今井 秀樹, 符号理論, 電子情報通信学会,pp.151–190,1990 年.

[4] Sophos, “Typosquatting-what happens when you mistype a website name?, ” http://nakedsecurity.sophos.com/typosquatting/