ステガノグラフィを用いたパスワードリマインダシステムの試作

宮元　章*・河野 哲也**・中川 竣太***・脇山 正博

Making prototype of a password reminder system using steganography

Akira MIYAMOTO, Tetsuya KAWANO, Shunta NAKAGAWA, Masahiro WAKIYAMA

Abstract

This paper discusses making a prototype of password reminder system using steganography. If you've forgotten your password, you are able to reset your password by password reminder. The password reminder is the function that is able to reset the password by the pre-registered answer and question. But, it is necessary to exactly match the answer and pre-registered answer. For example, if you answer the question “What is your favorite fruit?”, you can answer “lemon”, “LEMON”, “レモン” or “檸檬”. If “lemon” is the correct answer, “LEMON” , “レモン” and “檸檬” are wrong answers. Therefore, we propose a means to reset password by image file that is made by steganography. The means is to conceal the text information of pre-registered question and answer to the image file, then if user uploads the image to our developed system, user is able to reset password.

Keywords : steganography, password reminder

1. 緒言

電子メールやグループウェア等，様々な情報システムは，本人性の確認のためユーザ名とパスワードを入力し，認証を得られた際にそのシステムを利用することができる。利用するパスワードは，悪意ある第三者から解読され，アカウントを不正に利用されることを防ぐため，強いパスワード作成並びにパスワードの定期的な変更が必要である。情報処理推進機構が推奨する強いパスワードとは，数字，英字，記号をランダムに組み合わせた8文字以上のものが望ましいとされている[(1)]。しかし，上記のように適切なパスワード管理を徹底すればするほどパスワード失念の可能性を高めてしまう。それを回避するためにパスワードリマインダという仕組みが存在する。パスワードリマインダとは，パスワードを失念してしまった際，予め本人が登録した特定の質問に解答することで本人性を確認し，パスワードをリセットすることができる機能である。しかし，その答えは，完全一致するものでなければならない。例えば，好きな果物は？という問いに対し，「レモン」，「檸檬」，「lemon」，「LEMON」等様々な解答方法が存在する。また，はじめて海外旅行した年は？という問いに対し，「２０１４年」，「2014年」といった文字の全角/半角の違いであっても異なった解答となってしまう。

そこで本研究では，システムがいくつかの質問およびその解答を自動的に生成後，既存のパスワードリマインダシステムにバックグラウンドで接続しそれらを登録する。同時に，ステガノグラフィの技術を用い，それらの情報をユーザが用意した任意の画像ファイルに隠匿させ，ユーザはその画像ファイルを保管しておく。パスワード失念時には，そのファイルをシステムにアップロードすることでパスワードのリセットを行うことができる。このようなシステムを試作することを目的とする。

* 教育研究支援室 機器分析技術グループ
** 制御工学専攻 1年　　*** 制御情報工学科 5年

2. システム概要

2. 1. パスワードリマインダ UnifIDone

本校の所属する独立行政法人国立高等専門学校機構は，富士通株式会社製のUnifIDoneにより全国51高専の教職員・学生ユーザの一元管理を行っている[(2)]。各ユーザは自身のパスワードを自由に変更したりパスワード失念時にはパスワードリマインダの機能を利用し，パスワードのリセットを行ったりすることができる。

2. 2. 既存のパスワードリマインダへのバックエンド接続

既存システムのパスワードリマインダ機能では，無論，ステガノグラフィ技術を用いてパスワードをリセットすることはできない。そこで，本研究で試作するパスワードリマインダシステムは，新規に構築するサーバ上でステゴデータの作成やシークレットデータの抽出等のステガノグラフィに関わる処理を行い，バックエンドで既存のパスワードリマインダのサーバと双方向で連携し，既存のシステムを一切変更することなくステガノグラフィ技術を用いたパスワードリマインダ機能を実現した。

2. 3. パスワードリマインダへの登録

図1(a)にパスワードリマインダ登録時の流れを示す。パスワードリマインダの登録はすべてWebブラウザから行う。まず始めに ①ユーザ名とパスワードを入力し，ログインする。次に，②ユーザが選んだ任意の画像をアップロードする。その後，③ランダムな質問番号および30文字のランダムな文字列をその解答として自動生成する。④そのランダムな値および文字列をアップロードされた画像に埋め込む。また，⑤先ほどのランダムな値および文字列を既存のシステムに対しパスワードリマインダ登録を行う。最後に，⑥ユーザはそれらが埋め込まれた画像をダウンロードし，パスワード失念時に備える。また，図2に示す通り，ダウンロードした画像は

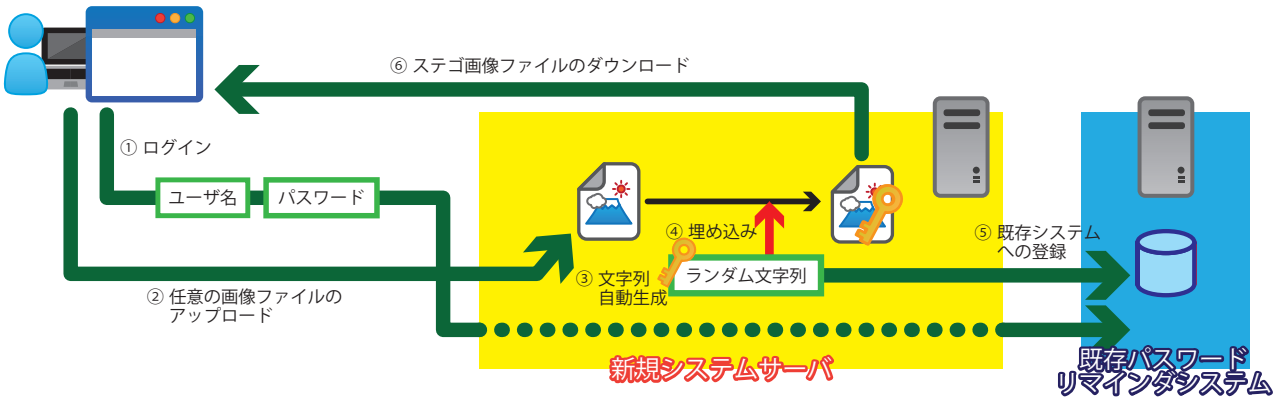

図1(a)　パスワードリマインダ登録時の流れ

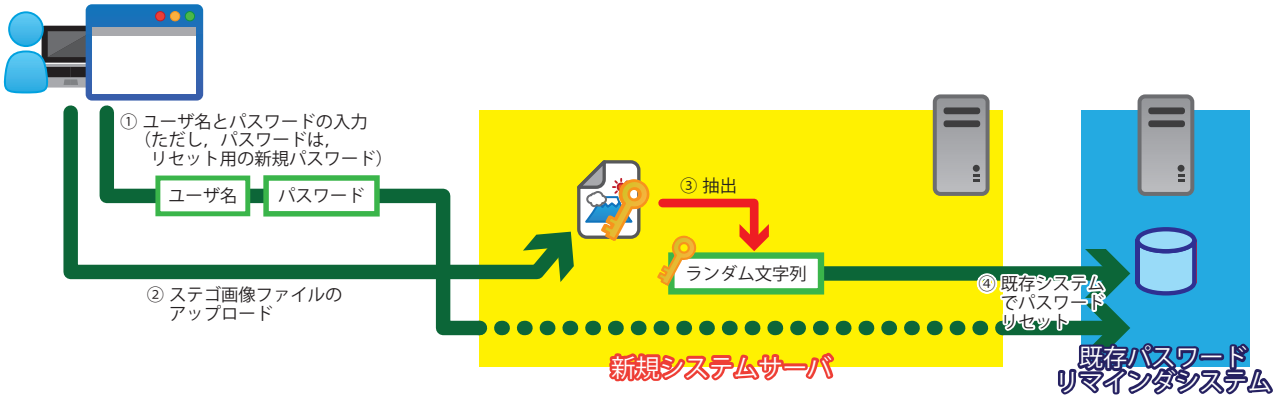

図1(b)　パスワードリセット時の流れ

他の画像と一緒に保管することで，第三者はどの画像ファイルにパスワードリマインダ情報が埋め込まれているかどうか判断することはできない。

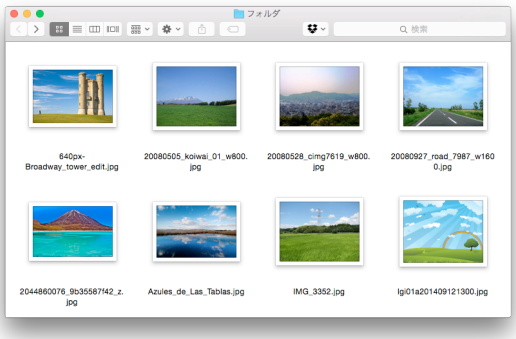

図2　どの画像に埋め込まれているか判断できない

2. 4. パスワードのリセット

図1(b)にパスワードリセット時の流れを示す。パスワードをリセットする際は，まず始めに ①ユーザ名，リセットするパスワードと共にパスワードリマインダ情報が埋め込まれた画像ファイルをアップロードする。次に，②その画像から質問番号並びに解答の文字列を抽出する。③抽出されたものを既存のリマインダシステムに送り，そのシステム上でパスワードをリセットする。

3. ステガノグラフィ

3. 1. ステガノグラフィ技術

ステガノグラフィ(Steganography)[3]とは，情報ハイディング技術の一つで，秘密にしたい情報を別のありふれたものに秘匿する技術である。そうすることで第三者からは秘密情報の存在自体を隠すことができる。

同じく情報ハイディングの1つであるクリプトグラフィ（Cryptography）と大きく異なる点は，クリプトグラフィは見ただけですぐに暗号化していることを悟られてしまうため，無尽蔵に時間をかけることで必ず解読され，秘密にしたい情報を保護することはできない。それに対し，ステガノグラフィは秘密情報が隠されていること自体を第三者に気付かせないようにするものであるため，秘密データを盗まれるという危険が極めて低いと言える。

一般的には，音声や画像などのデータ（カバーデータ）に秘密データ（シークレットデータ）を埋め込み，情報を秘匿したデータ（ステゴデータ）を作成する。このとき，カバーデータにもシークレットデータにも種類の制限はなく，画像，

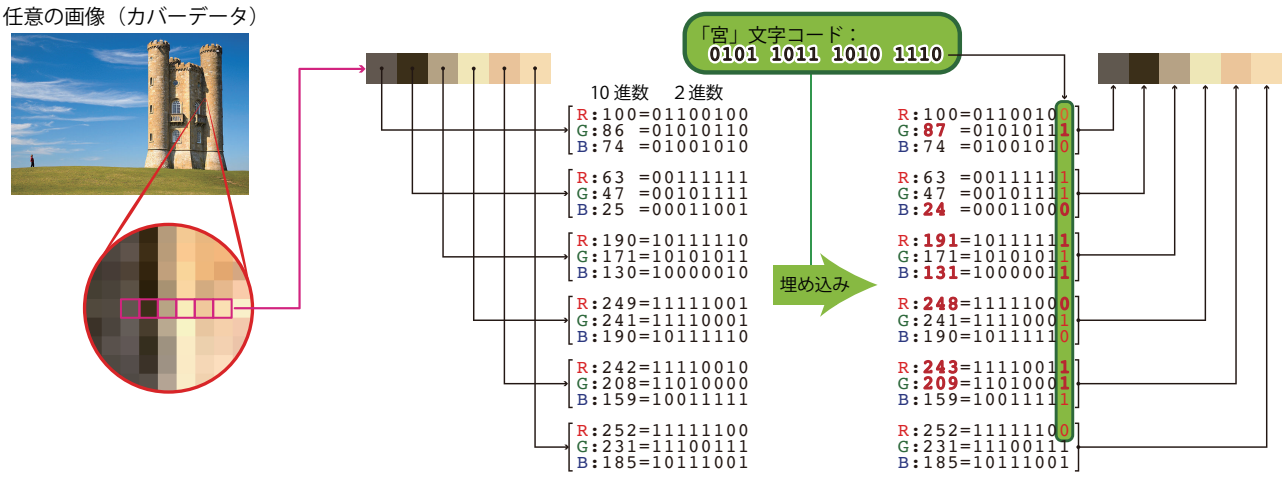

図3　画像ファイルの最下位ビット置換法による埋め込み例

表1　作成クラス表

クラス		メソッド	
名前	役割	名前	役割
KctSteganographyImage	画像を読み込む	readImageFile	画像を読み込む
		getBufferedImage	画像の読みとり
		getIntArray	画像を配列にする
KctSteganographyArray	RGBの配列	getRGBArray	配列をRGBそれぞれに分ける
		getR	Rの値を取得
		getG	Gの値を取得
		getB	Bの値を取得
KctSteganographyString	文字列変換	strToBinStr	文字コードから2進数の文字列を返す
		binStrToInrArray①	2進数の文字列配列を指定する桁数の整数型配列に変換
		binStrToIntArray②	2進数の文字列配列を16桁の整数型配列に変換
		strToIntArrayIncludeTerminateStr	埋め込み専用の文字列を整数型配列に変換
		intArrayToStr	文字列を数字に変換する
KctSteganographyEmbed	埋め込み	コンストラクタ	—
		getStegoRGBArrayByLSB	ステガノグラフィのRGB配列データを得る
KctSteganographyStruct	画像組み立て	コンストラクタ	—
		structImage	画像をRGBの配列から画像に戻す
		getImageFile	画像をRGBの配列から得た画像ファイルを得る
KctSteganographyExtract	抽出	コンストラクタ	—
		getSecretString	画像に埋め込まれた文字を抽出する

音声，動画，テキストなど，形式を問わず様々なものに埋め込むことが可能である。また，最下位ビットや離散コサイン変換，RAW画像の圧縮を用いるなど様々なアルゴリズムが考案されている。

3．2．最下位ビット置換法

最下位ビット置換法はステガノグラフィの埋め込みアルゴリズムの1つで，カバーデータの各単位の最下位ビットにシークレットデータのビット列を順に埋め込む方法である。

図3に示すようにカバーデータとして24ビット画像を例に取ると，画像の1ピクセルにあるR，G，Bの階調はそれぞれ8ビットで表すことができる。仮に「宮」という文字を埋め込む場合，その文字コード（Unicode）を2進数で表すと，”0101 1011 1010 1110”となる。文字コードの先頭から1ビットずつをR，G，Bの最下位ビットと置換することで埋め込みを行う。実際には複数の文字を埋め込む必要があるため，文字列の最後に終了文字を埋め込むことでシークレットデータを読み出す際の目印となる。最下位のビットのみの変更しか行われないため，データの劣化を少なくすることができる。実際に埋め込まれた画像は，元の画像と比べてもほとんど画像劣化を感じさせることはない。また，VGAサイズ（640ピクセル×480ピクセル）の画像をこの方法で埋め込む場合，終了文字を含めると57,600文字を埋め込むことができ，パスワードリマインダ情報として質問番号並びに数十桁の解答の文字列を埋め込むには十分である。

4. 開発方法

4. 1. 開発環境

サーバサイドプログラミングの開発言語としてJSP（Java Server Pages）及びServletを用いた。このことで，新たにJava言語で作成したクラスファイルの他に過去に作成したクラスファイルをプログラミング資産として利用することができ，効率よく開発を行うことができた。また，開発環境は，統合開発環境であるeclipseを用いた。また，動作環境は，OSをUbuntu Linux 14.04 LTS，Webサーバを Apache2.4，アプリケーションサーバをTomcat 7.0として構築した。しかし，Tomcatに付属するWebサーバは簡易的で実運用には不向きである。そこで，ApacheとTomcatを連携させることで安定稼働を図った。

4. 2. 開発方法

本研究では，数人で協力してプログラミングを行った。そのため，システムの計画段階において一連の流れを把握し，その流れを役割毎に幾つかの動作に分割し，その動作毎にJavaクラスを作成した。作成途中にメソッドの引数修正やメソッド追加・削除等必要に応じてクラスの内容を変更した。変更する内容を即座にプログラム作成者間で共有するため，表1に示すクラス作成表をクラウド上に保存し，いつでも閲覧できるようにすることで効率良くプログラミングを行うことができた。また，埋め込む方法として今回は最下位ビット置換法のみのクラスを作成したが，今後は離散コサイン変換，RAW画像の圧縮を用いた埋め込み方法のクラスを作成し，様々な埋め込み方法に対応する予定である。

4. 3. 研究室内でのJavaクラス共有

研究室では，毎年卒業研究生がステガノグラフィの研究を行っているが，その都度先輩のステガノグラフィに関わるソースコードを理解し，その後研究分野のコーディングを行うことになっている。しかし，それでは実際の研究に取り掛かるまで少なくとも数週間を要し，とても非効率である。そこで今後は，今回作成したJavaクラスをJavadocコマンドでクラス・メソッドに関するドキュメントを自動生成し，そのドキュメントをWeb上で閲覧することでソースコードを完全に理解することなく研究に取り掛かることができ，効率化を図ることができる。

5. パスワードリマインダシステムの操作例

図4(a)～図4(e)にパスワードリマインダシステムの操作画面例を示す。図4(a)にログイン画面を示す。この画面から，ログイン，パスワードリセットを行うことができる。ログイン後にはユーザが用意した任意の画像にパスワードリマインダ登録することができる。

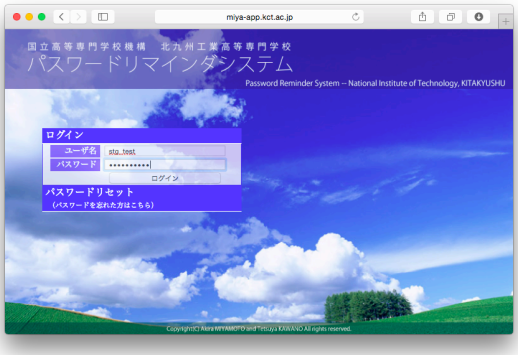

図4(a)　ログイン画面

図4(b)　パスワードリマインダー画像の登録

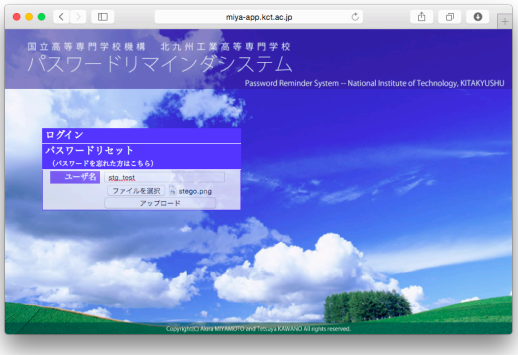

図4(c)　パスワードリセット

図4(d)　リセットするパスワードの入力

図4(e)　パスワードリセット成功

5. 1. パスワードリマインダ登録

ログイン直後の画面を図4(b)に示す。[ファイルの選択]をクリックし任意の画像ファイルを選択する。その後，[画像登録・ダウンロード]をクリックすると，そのファイルがサーバにアップロードされ，そこでパスワードリマインダの問題番号と解答の文字列を埋め込み，その画像ファイルをダウンロードすることができる。ダウンロードした画像はパスワード失念時に備え，ユーザが保管しておく。

5. 2. パスワードリセット

[パスワードリセット（パスワードを忘れた方はこちら）]を選択した画面を図4(c) に示す。パスワードリセットを行うユーザ名を入力し，問題番号と解答の文字列が埋め込まれた画像を選択後，[アップロード]をクリックしアップロードを行うと図4(d)に示す画面に遷移する。そこでリセットするパスワードを入力し，[リセット実行]をクリックするとパスワードリセットを行うことができる。パスワードリセットが成功した際の画面を図4(e)に示す。

6. 結言

本研究では，ステガノグラフィを用いたパスワードリマインダシステムの試作を行った。既存のパスワードリマインダシステムでは，パスワードをリセットする際，あらかじめ登録した質問内容並びにその解答を入力する必要があったが，本システムではユーザが用意した任意の画像を最下位ビット置換法によってそれらの情報を埋め込んだ画像を作成すし，その画像を用いることでパスワードリセットを行うことができるものとなった。

今後は，このシステムを複数の人に試験的に利用していただき，そこで出たご意見を参考にバグフィックス並びにマイナーチェンジを行った後，本校の学内ネットワーク内で教職員・学生の皆様に利用してもらえるよう本運用へと移行していきたいと考えている。

また，今回作成したJavaクラスはそれに関するドキュメントと共にAPIとして研究室内で共有することで今後は過去のソースファイルを完全に理解しない状態であっても効率的にステガノグラフィの研究を行うことができる。現在，Javaに限らずステガノグラフィのAPIがWebに公開されていることはほとんど無い。そこで，研究室内での共有が奏功した暁にはこのAPIを外部に公開することも視野に入れている。

参考文献

(1) IPA 独立行政法人 情報処理推進機構：コンピュータウイルス・不正アクセスの届出状況[2008年9月分および第3四半期]について
http://www.ipa.go.jp/security/txt/2008/10outline.html#5

(2) 国立高等専門学校機構様、全国51校の国立高等専門学校の認証基盤システムを統一 / 富士通
http://pr.fujitsu.com/jp/news/2012/07/17.html

(3) 脇山正博・田中義人・田中宏，情報セキュリティのステガノグラフィ，技術士 IPEJ Journal 2007年11月号，pp.8-11，2007

(2014年11月10日 受理)