情報処理学会論文誌 Vol.54 No.10 2300–2315 (Oct. 2013)

小型電子機器による情報の長さを考慮したポイゾニング手法

石井 順也[1,a)] ノールアフィザ マットラザリ[1,b)] 手塚 伸[2,c)] 宇田 隆哉[1,d)]
木下 俊之[1,e)]

受付日 2013年1月14日, 採録日 2013年7月13日

概要：近年，情報漏洩に関する問題が以前よりも注目されるようになってきている．情報漏洩の原因としては，ハッキングによる不正アクセスもあげられる一方，盗み見によるものや仕組みの弱点をつくものなどもあり，正当な権限を持たないものが情報にアクセスできないような技術自体を確立するのは困難であるといえる．そもそも情報は利用されるために存在しており，正当な利用者がある程度の利便性を持って利用できる情報は，不正な利用者にも利用される可能性を含んでいるからである．そこで，近年着目されているのが，情報をポイゾニングする手法である．情報が正確ではなくなれば，不正利用するものにとっての価値は下がる．しかしながら，情報自体がレイアウトを持つ状態として保持されている場合，そのレイアウトを崩さずなおかつ不自然にならないようにポイゾニングすることは困難である．そこで本論文では，オリジナルの情報が格納された文書のレイアウトを崩さず，なおかつポイゾニングしていることを気づかれず，一定以上の割合で不正確な情報を提供可能にするポイゾニング技術を提案する．本論文の提案技術を用いれば，情報漏洩自体を防ぐことができなかった場合においても，その被害を最小限に抑えることが可能となる．

キーワード：ポイゾニング，拡張現実感技術

Confidential Information Poisoning Methods by Considering the Information Length in Electronic Portable Devices

Junya Ishii[1,a)] Noor Afiza Mat Razali[1,b)] Shin Tezuka[2,c)] Ryuya Uda[1,d)]
Toshiyuki Kinoshita[1,e)]

Received: January 14, 2013, Accepted: July 13, 2013

Abstract: In recent years, problems arising from leakage of personal information become conspicuous. It is difficult to prevent malicious users from acquiring information since the method for the acquisition is not only hacking but also peeping. To begin with, information is worthy when it is referred to. Therefore, there is a possibility of acquiring information by malicious users, if the information can be acquired by correct users. On the other hand, information poisoning methods become popular. Information is depreciated for malicious users when its accuracy decreases. However, precise poisoning is difficult since space for words which are targets of poisoning is sometimes fixed, and unnatural space can be easily found by malicious users when the words are replaced with the candidates of words with different length. Therefore, in this paper, we propose a method for information poisoning where whole layouts of a document look natural and where appropriate candidates of words for poisoning are reserved by a fixed rate. The proposed method minimizes influence of leakage of information

Keywords: poisonig, augmented reality

[1] 東京工科大学大学院
Tokyo University of Technology, Hachioji, Tokyo 192–0982, Japan
[2] 慶應義塾大学大学院理工学研究科
Graduate School of Science and Technology, Keio University, Yokohama, Kanagawa 223–8522, Japan

[a)] ishii@u-lab.cs.teu.ac.jp
[b)] fiza110909@gmail.com
[c)] tezuka@mos.ics.keio.ac.jp
[d)] uda@cs.teu.ac.jp
[e)] kinoshi@cs.teu.ac.jp

1. はじめに

近年，インターネットの普及にともない，文書を電子的に扱う機会が増えてきている．とりわけ，スマートフォンの普及により，ビジネスにおいてもプライベートにおいても電子的な文書をいつでもどこでも閲覧できるようになりつつある．しかしながら，電子文書の利便性が情報漏洩の危険を増加させている．これは，正当な利用者がネットワークを通じて簡単にアクセス可能な電子文書は，不正な利用者にとっても同様であるからである．通常，第三者に公開されない電子文書にアクセスする際には，パスワードなどの保護手段が用いられている．しかし，複雑で安全なパスワードは正当な利用者の利便性を低下させ，利便性の高いパスワードは不正な利用者にとっても解読しやすいものとなってしまう．電子文書へのアクセス保護手段としては，利用者の生体情報を用いたものや特殊な認証デバイスを用いたものも存在するが，一般に普及するほどコストが安価ではない．

そこで近年注目されてきているのが，情報をポイゾニング [1], [2] によって保護する手法である．

Lin らによる研究 [1] では，プライバシに関する情報や，違法なコンテンツや，他人の著作物が，BitTorrent システムの中の DHT（Distributed Hash Table）ネットワークを流れるのを，ポイゾニングによって妨害する手法を提案している．この手法の特徴は，ネットワーク全体を使えないようにするのではなく，特定の通信のみを妨害できる点にある．この研究では，第三者が故意に流出させている情報を，ポイゾニングによって妨害することで流出させないようにしている．

また，Ortega らによる研究 [2] では，ARP（Address Resolution Protocol）のキャッシュが，攻撃者にポイゾニングされないようにする対策手法が述べられている．この場合，ポイゾニングを行うのは第三者である攻撃者のほうであり，正しい情報を不正な情報に書き換えることで，他の利用者に誤った情報を与えている．

このように，ポイゾニングは，攻撃者が情報を汚染する手段としても利用できる一方，逆に，攻撃者に与えられる情報を汚染することで攻撃を防ぐ手段としても利用できる．本提案手法もポイゾニングを利用しており，その手法は後者に属すが，情報を提供する主体とポイゾニングを行う主体が同一である．さらに，ポイゾニングに使用する情報が，オリジナルの情報に対して違和感がないことが本提案手法の特徴である．

ポイゾニングにより，不正な利用者が入手する情報が不正確になればその価値は大幅に低下する．たとえば，不正に入手された社内情報の確度[*1]が 0.01%であった場合，その情報の利用価値はきわめて低いであろうし，同様の確度のプライバシ情報であればことさらである．紙面ではない電子媒体は，書き換えや偽造の痕跡を残さずに情報のポイゾニングを行うことが容易であるため，ポイゾニングによる情報の保護手法は情報漏洩に対して有効であるといえる．しかしながら，電子的な情報であっても，書き換えや偽造の痕跡を残さずに自在に任意の値に書き換えられるとは限らない場合がある．

近年，コンピュータを取り巻く環境が急速に発展したため，情報が文字データという単純な形で扱われることは稀となり，装飾が施された見栄えの良い文書形式であることが多い．そのような形式は必ず文書のレイアウトをともなっており，一般的にはそのレイアウトが美しくなるように文書が作成されている．このとき，情報をポイゾニングしようとして特定の文字列を異なる任意の文字列に置き換えてしまうと，レイアウトが崩れてしまうことがある．たとえば，オリジナルの文書では 4 文字で記載されていた情報に対し，10 文字の情報でポイゾニングした場合，不自然さをともなうことなくこの文書を表示することはほとんど不可能である．また，適切なレイアウトとなるようにのみ注意を払い，ランダムな項目でポイゾニングを行うと，内容が不自然になることもある．たとえば，関東地方の地名欄に九州の地名が書き込まれていれば，故意に改変された情報だと気づかれやすくなってしまう．さらに，任意の項目に対してポイゾニング候補となる項目が少ない場合，情報の確度が上がってしまう．

以上の問題点を考慮し，本論文では，電子的な文書に対して適切なポイゾニングを行う技術の提案を行う．

本論文では，2 章にて関連研究とその問題点を述べ，3 章にて提案手法を説明し，4 章にて実装について述べ，5 章にて実装に関する実験を説明し，6 章にて実験結果を考察する．最後に，7 章にて本論文をまとめる．

2. 関連研究

2.1 紙媒体の暗号化

紙媒体における暗号化技術として，阿南らによる紙の暗号化技術 [3] がある．紙媒体の暗号化は，印刷物を電子データのように暗号化し，パスワードを知る人のみが隠蔽された情報を閲覧することができる技術である．紙の暗号化は，暗号化したい領域を画像として認識し，画像のスクランブル処理などをし，暗号化を行うものである．そして，暗号化された画像を紙に印刷する．暗号化された領域を復号する場合は，イメージスキャナやカメラなどの画像処理デバイスで暗号化領域を読み込み，復号用ソフトウェアによって復号を行う．そのため，復号するためには復号用のソフトウェアとパスワードの両方が必要であり，第三者には暗号化された情報を復号することができないため，暗号化された情報が外部に流出することがない．

[*1] 本論文では，情報の確からしさの割合を「確度」という言葉で表現する．提示された情報が 50%の確率で正しいとき，その確度は 50%である．

しかし，ソフトウェアを所有している組織内で，暗号化された書類を見る権限のない人間が，書類を閲覧しようとした場合，パスワードが 4 桁の数字の場合であれば 10,000 通りしかないため，総当たりでパスワードを解くことが可能である．画像処理デバイスに入力するパスワードの桁数を増やすなどの対策をしても，ユーザがパスワードを忘れてしまう恐れや，入力自体に負担がかかってしまうといった問題点がある．

2.2 AR を用いたアクセス制御可能な紙広告媒体の提案

紙媒体におけるアクセス制御の手法として，渡部らの AR を用いたアクセス制御可能な紙広告媒体の提案 [4] がある．これは紙広告において，掲載情報をネットワーク上のサーバから取得し，取得した情報を紙に印刷されているかのように表示することで，個人別の情報表示および，配布後の掲載内容の変更を実現している．AR の表示に用いるマーカを 2 次元コードにし，2 次元コードに含まれている広告 ID とユーザ自身の ID をサーバに送ることにより，AR で表示されるデータが個人ごとに区別されるようになっている．

2.3 P2P ネットワークにおけるポイゾニング手法

ポイゾニングの手法の中に，P2P ネットワークにおけるインデックスポイゾニング [5], [6], [7], [8], [9], [10] がある．

インデックスポイゾニングとは，ファイルの流通制御を行うために加工したダミーファイルキーをネットワーク上に拡散する手法である．拡散されたダミーファイルキーは，元ファイル名と同じにする．これにより，ノードが検索をかけ，ダミーファイルキーを入手した場合，入手したファイルキーはダミーであるため架空のファイルとなる．これにより，拡散させたくない元のファイルがダウンロードされる確率が低くなる．

ポイゾニングを行う利点として，本物のファイルを，偽装した大量のダミーファイルで隠蔽し，本物のファイルの拡散を制御できるという点がある．

本論文では，このポイゾニングという技術を用いて，元データと違和感のない偽情報を攻撃者に閲覧させることにより，元データの特定を困難にさせる手法をとる．

2.4 ブルームフィルタを用いたプライバシ保護検索における攻撃モデルとデータ撹乱法の一検討

渡辺らのブルームフィルタを用いたプライバシ保護検索における攻撃モデルとデータ撹乱法の一検討 [11] においてもポイゾニングによるプライバシ情報の保護がなされている．この手法では，タプルの内容が暗号化されているが，複数のタプルが同じ値を持つ場合，それらのタプルの内容は暗号化されていたとしても同じビットパターンになってしまうため，同一のタプルを高い頻度で使用すると，テーブルの内容が攻撃者に推測されやすいという問題がある．そこで，正しい情報のビットのマップに適切なノイズを加味することで，攻撃者にテーブルの内容を推測させないようにしている．

この手法からも，ポイゾニングにより，偽情報によりオリジナル情報の確度を下げ，オリジナル情報の保護を行うことは可能であり，有効であると考えられる．

3. 提案手法

本提案で留意する点は，書類のレイアウトを崩さないことと，ポイゾニングに使用した情報が，オリジナルの情報に対して違和感がないということ，ポイゾニングにより，文書の情報の確度を 0.01%以下にする（理由は 3.2 節にて後述する）という 3 点である．

レイアウトの崩れとして，たとえば，AR によって表示されるポイゾニングを行った偽情報がオリジナル情報に比べ極端に長くなると，すでに印刷されている情報に重なってしまうといったことがあげられる．この場合，攻撃者に違和感を与えてしまう．このポイゾニング情報の長さのほかにも，攻撃者に違和感を与えてしまう要因として，ポイゾニングに使用した情報がオリジナルの情報に対して関連性が低くなるということが考えられる．使用者に違和感のある情報を表示した場合，情報の確度が上がってしまい，オリジナルの情報やパスワードの特定につながる恐れがある．

オリジナルの情報の確度を 0.01%以下にするため，閲覧用ソフトウェアで使用されるパスワードは数字 4 桁にする必要がある．そして，ユーザが閲覧用ソフトウェアに誤ったパスワードを入力しても認証されているかのように，ポイゾニングされた偽情報をデータとしてユーザに返送することで，オリジナル情報の特定を防ぐ．そのため，ポイゾニングに使用される偽情報はパスワード 1 つに対し 1 つ必要となり，ポイゾニング候補の総数 10,000 通り以上必要となる．

なお，本論文では，スマートフォンなどの携帯デバイスから情報にアクセスすることを想定している．文書の本体はスマートフォン上にあっても，ネットワークで接続されたサーバ上にあってもよいが，秘匿にする情報だけはサーバにのみ保存され，ファイルを開く際に端末に送信される．このとき，端末のロック解除時に入力されたパスワードに対応して，端末に送信される情報がサーバ上で決定される．正しいパスワードであれば正しい情報が送信され，誤ったパスワードであればポイゾニングされた情報が送信されるが，正しいパスワードを知らない不正ユーザには，入力されたパスワードに応じて表示される情報のうち，どの情報が正しいものであるのか区別が付かないというのが本手法の特徴である．

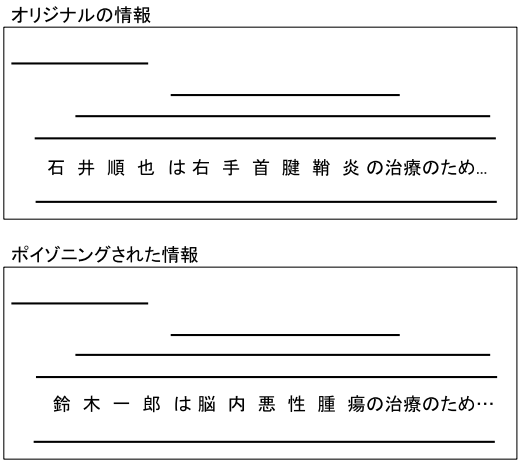


図 1 違和感を覚えさせないポイゾニング手法の概要
Fig. 1 Overview of poisoning method that attarcts little attention.

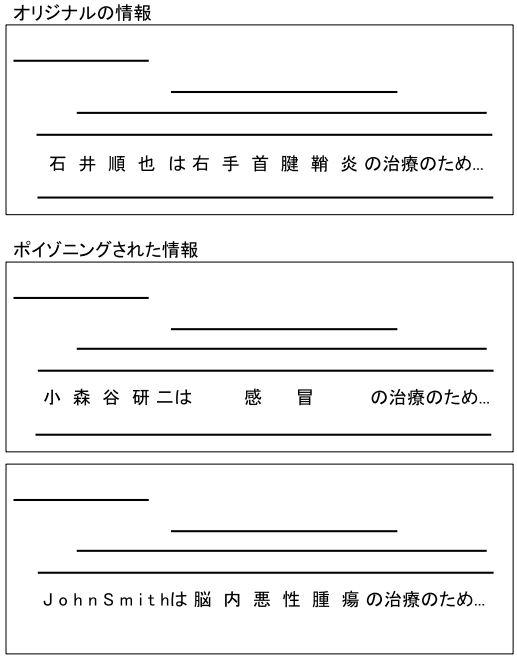


図 2 違和感を覚えるポイゾニング
Fig. 2 Poisoning method that attarcts considerable attention.

3.1 文書のレイアウトと違和感

文書の発行を行う制作者は，文書上で保護したい箇所を選択し，その情報が何であるのか種類を決定する．選択した箇所は文書中から削除され，削除された内容を置換するものを入れるために，オリジナルの情報（保護情報）の 2 倍の文字数となるスペースが確保される．保護情報のみをスペースに置き換えることで，保護情報以外の文章のレイアウトは崩れないため，閲覧者は文書をストレスなく読むことが可能である．

上記にも述べたとおり，保護情報であるオリジナルの情報の長さの 2 倍のスペースが確保されているため，ポイゾニングに使用する情報は，この文字数の範囲内で選び出す必要がある．ポイゾニングに使用する情報がランダムに選び出された場合，オリジナルの情報と比較し文字数が極端に長い場合保護情報以外の文章に重なってしまいレイアウトが崩れる．極端に短い場合は違和感を与えてしまいポイゾニングが適切に行えないことが考えられる．文字数以外に違和感を与えうるのは，オリジナル情報とポイゾニング情報の関連性が低い場合である．たとえば，オリジナル情報が日本語名にもかかわらず，ポイゾニング情報は海外名であった場合，攻撃者は，保護情報以外の文脈からポイゾニング情報を不自然に感じてしまう．

以上より，ポイゾニングに使用する情報は，レイアウトを崩すことなく，オリジナル情報と比較しても自然である必要があるといえる．

本論文で提案するポイゾニング手法を**図 1** に示す．正規のユーザが情報にアクセスする場合，図中のオリジナルの情報がそのまま表示される．一方で，不正なユーザが情報にアクセスする場合には，図中の保護情報に指定された情報のみが，ランダムに選ばれた他の情報に置換されて表示される．置換対象として選ばれる候補は，不正ユーザに違和感を与えないよう，レイアウトおよび内容が考慮されている．

なお，3.2 節にて詳述するが，違和感を覚えさせないために本論文で考慮しているのは，レイアウトと意味的な内容の 2 点である．**図 2** に違和感を覚えるポイゾニングの例をあげ，理由を示す．

図 2 の 1 つ目に示されているポイゾニングされた情報に関しては，表示されている病名の候補の中に明らかに文字数が少ない「感冒」というものがある場合を例示するものであり，これがレイアウトに関する違和感である．同図の 2 つ目に示されているポイゾニングされた情報に関しては，患者名がどの候補も日本人名ばかりである中に明らかに日本人名ではない「John Smith」というものがある場合を例示するものであり，これが意味的な内容に関する違和感である．本提案手法が実現するのは，レイアウトに対しても意味的な内容に対しても攻撃者に違和感を覚えさせないようにするポイゾニングである．

3.2 ポイゾニング手法

3 章冒頭でも述べたとおり，ポイゾニングされた情報が，オリジナルの情報に対して違和感がないことと，ポイゾニングされた情報の確度が低いこととが重要である．本論文では，確度が 0.01%以下となるようにポイゾニングを行っている．銀行の ATM では，利便性と安全性の両面を考慮し，数字 4 桁を暗証番号としている．つまり，社会通念上安全と見なされているパスワードは，最低数字 4 桁であると考えられる．クレジットカードを利用する際に求められる暗証番号も数字 4 桁であり，一般的な携帯電話の機能をロックする暗証番号も数字 4 桁である．本論文が想定して

いる情報の対象は，主にスマートフォンなどで閲覧可能なものであり，スマートフォンも一般的に数字 4 桁のパスワードでロックされるようになっていることから，本論文でも数字 4 桁でロックが解除されることを想定している．表示された情報がポイゾニングされたものであるかどうかを知られないためには，入力された同一の数字に対して，同一の情報が表示される必要がある．入力される数字 4 桁に対応する情報は，全部で 10,000 通りであり，不正ユーザがランダムに数字 4 桁を入力して情報を閲覧する場合，その情報の確度は 0.01%になる．

これらの条件を満たすためのポイゾニング手法を，数値について 3.2.1 項で，人名，商品名，病名について 3.2.2 項で，地域，組織名について 3.2.3 項でそれぞれ述べる．

ここで，ポイゾニング対象として選択されている，3.2.1 項の数値，3.2.2 項の人名，商品名，病名，3.2.3 項の地域，組織名に関しては，本提案手法がポイゾニング対象として限定しているわけではなく，あらゆる語句を指定可能である．3.1 節で述べたように，どの情報をポイゾニングするのが有用かに関しては，文書の発行を行う制作者の判断に委ねられている．ただし，レイアウトおよび内容に対する違和感を不正ユーザに与えないようにするため，本論文では，3.2.1 項，3.2.2 項，3.2.3 項で取り上げられるような 3 つの種別に分類してポイゾニング手法を提案している．

また，どの情報をどの程度ポイゾニングするのが有効であるのかに関しては，前述した，確度を 0.01%以下にする理由と密接な関係がある．本論文が想定している環境では，不正ユーザが入力するパスワードが数字 4 桁であるため，10,000 通りの保護したい語句それぞれに対して，ポイゾニング候補も 10,000 通り用意可能である．それでは，10,000 通り以下のポイゾニング候補では安全性に関して有効ではないのかというと，それは不正ユーザがどの程度の知識があり，何をするのかにもよるため，その判断は容易ではない．

そこで，本提案手法では，安全性を優先し，不正ユーザに違和感を与えないポイゾニング候補がつねに最大限の 10,000 通りとなるようにしている．3.2.3 項で後述する地域名などの場合，最良の条件では 10,000 通りのポイゾニング候補を確保できない場合もあるが，その場合にも最大限用意可能な組合せでポイゾニングを行う．よって，より少ないポイゾニング候補を提示しても安全性に関する有効性が変わらない場合があったとしても，安全性に関する客観的な評価を省略し，安全性が最大になるよう，つねに最大限のポイゾニングを行うのが本手法の特徴である．

なお，本提案手法は，文書中のどの箇所をポイゾニングしているかを秘匿にするものではなく，秘匿にしたい情報の確度を 0.01%以下にすることで，攻撃者にとって得られる情報を価値の低いものにするものである．攻撃者がパスワードを 2 回入力すれば，同一箇所に 2 つの値を持つ文書が得られる．ポイゾニングされているのはその箇所であることが攻撃者には分かるが，その情報の確度が低ければ価値も低くなる．また，表示されている情報がポイゾニングされたものであるかどうか，文脈から攻撃者に推測されてしまうのではないかという懸念が考えられる．この点に関しては，3.1 節冒頭で述べたように，文書の発行を行う制作者が，文書上で保護したい箇所を適切に選択しているかどうかに安全性が依存する．たとえば，「八王子市にある東京工科大学」という文の，「八王子市」のみしか文書上で保護したい箇所として選択されなかった場合，「三鷹市にある東京工科大学」というポイゾニングされた文が作成されていたとしても，東京工科大学が三鷹市にないことを攻撃者が知っている場合には，この文がポイゾニングされたものであることを見破られてしまう．よって，文書の発行を行う制作者は，「三鷹市」という地域名と，「東京工科大学」という組織名の両方を文書上で保護したい箇所として登録しておくべきだったといえる．このように，文書の発行を行う制作者が，文書の構造を熟考した上で保護したい箇所を適切に選択していれば，提案手法に問題はないと考えられる．ただし，文書の発行を行う制作者が不適切な選択をした場合には，どの情報がポイゾニングされたものであるか，文脈から攻撃者に悟られてしまう場合もあることに留意されたい．

3.2.1 数値に対するポイゾニング

金額と会員番号の 2 つに分けて説明する．

まず，金額と区分された数値について述べる．攻撃者が，ポイゾニング箇所の前後の文章構成により，LSD（least significant digit）から数えた 0 の数が 3 桁の数値が正解であると推測されてしまったとする．その場合，8137，4387，2371，3893，3000 のポイゾニング候補から，5 回目の金額が正解であると推測され，パスワードが絞り込まれてしまう．したがって，ポイゾニングの候補は，元データの LSD から数えた 0 の桁数に合わせる必要がある．さらに，ポイゾニング候補を 10,000 通りにするには，0 から 9999 の数字となるため，ポイゾニング候補の桁数は最低 4 桁必要になる．しかし，元データの MSD（most significant digit）から数えた 0 以外の数字の桁数が 4 桁未満の場合，ポイゾニングの候補を 10,000 通り以上算出することができない．そこで，元データとポイゾニング候補の桁数合わせと，元データが 4 桁未満であった場合にポイゾニング候補を算出する手法を説明する．桁合わせが必要な数値算出のフローを**図 3** に示す．まず，元データの 10^0 の場所から，i の場所の 0 の数を，count をインクリメントし数える．次に，元データの LSD から MSD のビット数 n から *count* を引く．その値が 4 より少なかった場合，ポイゾニング候補は 10,000 通り以下になる．そのため，ポイゾニング候補を 10,000 通りにするために最大桁数を拡張する必要がある．式 (1) によってポイゾニング候補を算出する．

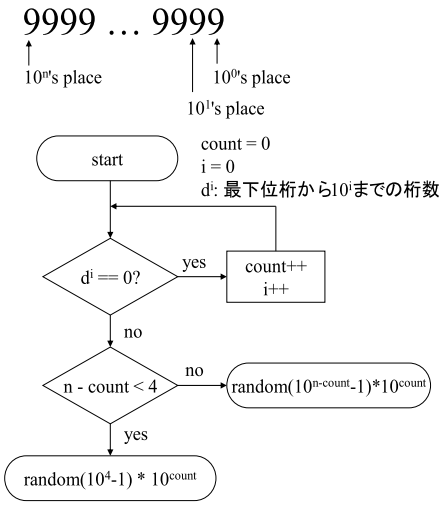


図 3 桁合せを必要とする数値のフロー
Fig. 3 Process flow of integers with fixed-length zeros.

$$Answer = random(10^4 - 1) * 10^{count} \quad (1)$$

$Answer$ はポイゾニング用の数値を表している．$random()$ は，最小値 1 から，括弧内の数式より求められる値を最大値としたランダムな数値を算出する関数である．最大値を 9999 に設定することにより，ポイゾニング候補は 10,000 通り確保することが可能である．$random$ 関数により算出された値に，10^{count} をかけ合わせることによって，元データの LSD からの 0 の桁数を合わせることができる．元データの LSD から MSD の桁数から count を減算した値が 4 より多かった場合は，10,000 通り以上のポイゾニング候補を得ることが可能である．そのため，(2) を用いてポイゾニング候補を算出する．

$$Answer = random(10^{n-count} - 1) * 10^{count} \quad (2)$$

式 (2) で算出した $random$ 値も元データと桁数を合わせる必要があるため，10^{count} をかけ合わせる．次に，ユーザの会員番号などのシリアル番号と区分された数値であった場合について論ずる．会員番号は会員それぞれに固有の番号がランダムに振り分けられているものであるため，金額のようにポイゾニング候補を元データの桁数に合わせると不自然になってしまう．そのため，会員番号は金額などの数値と違い，0 の桁数を合わせる必要がない．桁合わせの必要のない場合のフローを**図 4** に示す．まず，元データの 10^0 の桁数を数える．LSD から MSD の桁数が 4 より少ない場合，ポイゾニング候補は 10,000 通り以下になってしまう．そのため，$random$ によって算出されるランダム値の上限を 9999 にしなければならない．そこで式 (3) を用いてポイゾニング候補を算出する．

$$Answer = random(10^4 - 1) \quad (3)$$

LSD から MSD までの桁数が 4 以上であった場合は，ポイゾニング候補は 10,000 通り以上算出することができる．

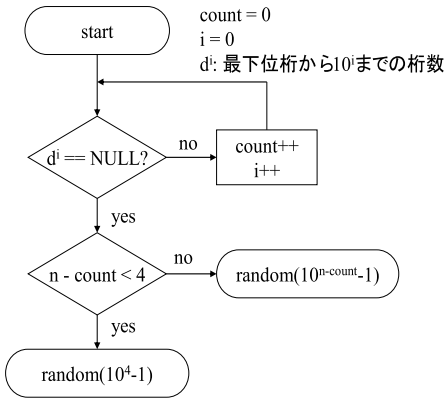


図 4 桁合せを必要としない数値のフロー
Fig. 4 Process flow of integers without fixed-length zeros.

LSD から MSD までの桁数が 4 以上であった場合は，ポイゾニング候補の算出には式 (4) を用いる．

$$Answer = random(10^{count} - 1) \quad (4)$$

以上で述べた手法であれば，元データが数値だった場合，10,000 通り以上のポイゾニング候補を算出することが可能である．さらに，攻撃者のパスワードをシードにした乱数を返すため，同一のパスワードを入力した場合表示される数値も同一のものになり，正解パスワードがどれかの判断ができなくなる．

3.2.2 人名，商品名，病名に対するポイゾニング

本項のポイゾニング手法は，ポイゾニング単語候補の検索を行う検索フェーズと，ポイゾニング単語候補の中からポイゾニング単語を採用する採用フェーズに分かれている．

人名，商品名，病名のポイゾニング候補は元データとの関連性が重要である．攻撃者が適当なパスワードを入力し次のような候補が得られたとする．Tanaka Jiro，Suzuki Ichiro，James Doe，Sato Saburo．この場合，攻撃者は得られた候補の中に英語名が 1 つしかないため，James Doe が元データだと推測することができ，パスワードが特定されてしまう．

そこで，元データと関連性の高いポイゾニング候補を選出する手法を説明する．

図 5 に，ポイゾニング候補を検索するフローを示す．$param$ は，テーブルに保存されている人名や商品名，病名の区分用のカラムである．データベースのテーブルに保存されているデータが属している $param$ には 1 のビットが格納されており，データが属していない $param$ には 0 が格納されている．人名の場合，$param1$ は日本人，$param2$ はアメリカ人とし該当する箇所に 1 が格納され，該当しない箇所には 0 が格納される．

元データからポイゾニング候補を選び出す条件は 2 つある．1 つ目は元データである $original\ word$ とポイゾニング用データ $poisoning\ word$ との文字数差の範囲である．2 つ目は $original\ word$ に紐付けられている $param$ の値と，

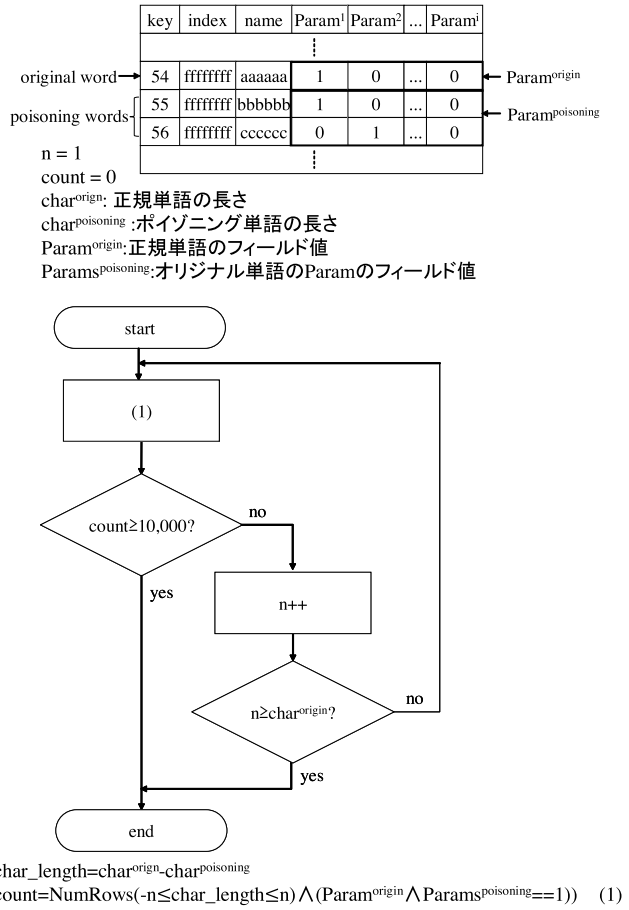


図 5 人名，商品名，病名に対するポイゾニング単語候補検索フロー
Fig. 5 Search flow of candidates for poisoning of names of people, products and diseases.

データベースのテーブルに保存されている *poisoning word* に紐付けされている *param* の値の論理和が 1 になるという条件である．以上の 2 つの条件に当てはまるポイゾニング候補を検索するために式 (5) を用いる．

$$count = NumRows(-n \leq char\_length \leq n \wedge (param_{origin} \wedge params_{poisoning} == 1)) \quad (5)$$

$$char\_length = char_{origin} - char_{poisonig} \quad (6)$$

式 (5) の $NumRows()$ は 2 つの条件式に当てはまる候補総数を返す関数を表している．$\wedge$ は論理和を表している．$char\_length$ は式 (6) を用いて算出する．$NumRows()$ から返ってきた値は *count* に代入される．n は *original word* と *poisoning word* との文字数の差の範囲を表しており，ポイゾニング候補の初回検索時は n の値は 0 である．式 (6) の $char_{origin}$ は *original word* の文字数を表している．$char_{poisonig}$ は *poisoning word* の文字数を表している．ポイゾニング候補の初回検索時に count が 10,000 通り以上になった場合はポイゾニング候補の検索を終了し，ポイゾニング候補の採用フェーズに移る．ポイゾニング候補の初回検索時にポイゾニング候補が 10,000 通り未満であった場合は，n をインクリメントし，*original word* と *poisoning word* との文字数の差の範囲を拡張する．n をインクリメント後，ポイゾニング候補が保存されているテーブル内を $NumRows$ 関数で再度検索を行う．$NumRows$ 関数で再検索後に，ポイゾニング候補が 10,000 通り見つからない場合は，*original word* と *poisoning word* との文字数の差の範囲を拡張し検索を繰り返す．文字数の差の範囲に用いている n 値の最大数は，*original word* の文字数とする．n が最大値になった時点で，ポイゾニング候補が 10,000 通り未満の場合であってもポイゾニング候補の検索を終了する．ポイゾニング候補が 10,000 通り未満だった場合は，ポイゾニング候補の検索終了時に見つかったものを使用しポイゾニング採用フェーズに移る．ポイゾニング単語の採用フローを**図 6** に示す．まず，単語採用キーである $Search\_key$ の算出を式 (7) で行う．

$$Search\_key = Hash(UsrRandNum||Pass) \quad (7)$$

$UsrRandNum$ は 160 bit ランダムのユーザ専用乱数を表している．Pass は攻撃者が入力したパスワードを表している．|| は右の値と左の値の連接を表している．$Hash()$ は，$UsrRandNum$ と $Pass$ を連接した値を SHA-1 によって 160 bit のハッシュ化させる関数を表している．式 (7) によって算出した $Search\_key$ を使用し，ポイゾニング単語の採用を行う．候補検索フェーズで選択されたポイゾニング候補の *index* フィールドの値と $Search\_key$ との一致率を算出する．ポイゾニング候補の *index* と $Search\_key$ との一致率は式 (8) によって算出を行う．

$$Match\_Rate = MATCH(Search\_key, index_{pois}) \quad (8)$$

$MATCH()$ は $Search\_key$ と $index_{pois}$ の一致率を算出する関数を表している．$Match\_Rate$ は，ポイゾニング候補の *index* と $Search\_key$ との一致率を表している．$index_{pois}$ はポイゾニング候補の *index* フィールドの値を表している．式 (8) によって求められた $Match\_Rate$ の中から，最も 1 に近い一致率のポイゾニング候補を，ポイゾニング採用単語として選び出す．ポイゾニング採用単語の選び出しは式 (9) によって行う．

$$Apword = (Match\_Rate \approx 1) \quad (9)$$

$Apword$ はポイゾニング採用単語を表している．$\approx$ は左辺の値が右辺の値に 1 番近い値のポイゾニング採用単語の選出を表している．ポイゾニング採用単語が 2 個以上あった場合，ポイゾニング採用単語に紐付けられている key が，最も 0 に近いポイゾニング採用単語を採用する．ポイゾニング採用単語が 2 個以上あった場合のポイゾニング採用単語の採用を行う式を式 (10) に示す．

$$Apword = min(Apwordkey_1, \ldots, Apwordkey_i) \quad (10)$$

index^{pois}

	key	index(160bit)	name	Param^1	Param^2	...	Param^i
Apwordkey^1 →	1	ffffffff	aaaaaa	1	0	...	0
Apwordkey^2 →	2	ffffffff	bbbbbb	1	0	...	0
	3	ffffffff	cccccc	0	1	...	0
	⋮						
Apwordkey^i →	i	ffffffff	zzzzzz	0	1	...	0

count = 0
i=1
Search_key : UsrRandNumとPassから得られるハッシュ値
Match_Rate: Search_keyとindex値の一致率
Apword : 採用されたポイゾニング用単語

Search_key=Hash(UserRandNum||pass) (1)
Match_Rate=MATCH(Serch_key,index^{pois}) (2)
Apword=(Match_Rate≈1) (3)
count=NumRows(Apwword) (4)
Apword=min(Apwordkey^1 , … , Apwordkey^i) (5)

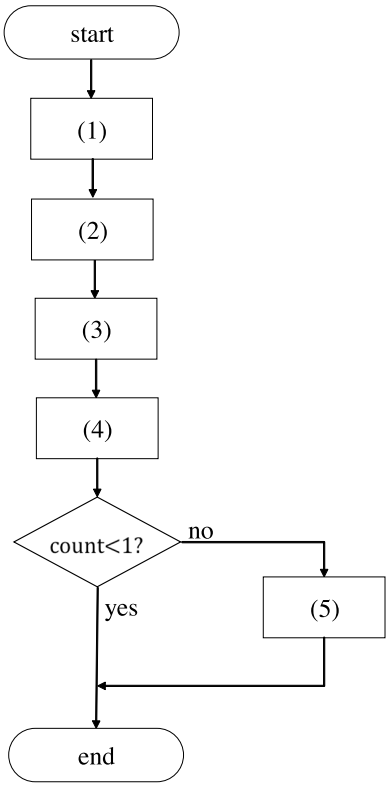


図 6 ポイゾニング単語採用フロー
Fig. 6 Selection flow of words for poisoning.

$Apwordkey_1$ は $Apword$ に紐付けられた key の値を表している．$min()$ は $Apword$ の key の値が 1 から i までの範囲で 0 に最も近い $Apword$ を抜き出す関数である．以上によりポイゾニング採用単語の選び出しが可能である．データベースの $param$ を人名，商品名，病名用に区分を書き換えることにより，人名，商品名，病名それぞれのポイゾニング候補の検索および，ポイゾニング単語の採用が可能である．

$UsrRandNum$ はユーザごとランダムに変化するため，データベースが更新されない限り，同じ偽のパスワードに対し，必ず同じポイゾニング単語が適切に選択される．データベース更新時に追加される新規データの量は，元のデータ量に対して極端に小さいと考えられる．そのため，データベース更新後に，ポイゾニング単語がデータベース更新前のポイゾニング単語と変わる場合はごくわずかである．さらに，データベース更新の前後でデータベースに大量にアクセスされて結果を大量に収集されない限り，AR によって表示されるデータがポイゾニングされた偽物であることを攻撃者に気づかれることはない．

3.2.3 地域，組織名に対するポイゾニング

地域のポイゾニング候補は，県，市，町，の 3 つにテーブルを分けて選出する．県名の場合も，人名と同じく元データとポイゾニング候補との関連性を高くさせる必要がある．元データと関連性の高いポイゾニング候補を選出するために，元データと同一の param に 1 が立っている県名をポイゾニング候補とする．県数は人名とは違い，実際に存在している数に限りがあり，さらに，国によっては 10,000 通り県が存在しない場合もある．したがって，人名のように文字数を拡張し，候補の検索範囲を広げても県名のポイゾニング候補は存在している県数以上増加しない．県名については同一 param に 1 のフラグが立っている県のすべてをポイゾニング候補とする．ポイゾニング候補内からのポイゾニング単語の採用方法は，人名と同様に，図 6 のフローを用いて，式 (7) から求められる $Search\_key$ とポイゾニング候補の index との一致率が最も高いポイゾニング候補をポイゾニング単語として採用する．

市名，町名，組織名のポイゾニング候補検索フェーズについて論じる．市，町，組織名のポイゾニング候補は，元データの地点から一定の範囲内のものを選出する．ポイゾニング候補検索フェーズを**図 7** に示す．ポイゾニング候補の検索条件は，*original word* と *poisoning word* との文字数差の範囲であることと，*original word* と *poisoning word* との距離が指定範囲内であるかということである．*original word* と *poisoning word* との文字数差の範囲は *original word*−5 文字から *original word* + 5 文字までの範囲である．2 つの条件に一致したポイゾニング候補の検索を行う際に用いる式は (11) である．

$$count = NumRows(-5 \leq char\_length \leq 5 \wedge dist \leq n \wedge (param_{origin} \wedge params_{poisoning} == 1)) \quad (11)$$

$$char\_length = char_{origin} - char_{poisoning} \quad (12)$$

条件に一致した *poisoning word* は $NumRows$ 関数によって個数を算出され，*count* に代入される．n 値は *original word* と *poisoning word* との距離の範囲を示しており初期値は 10 である．そのため，*original word* と *poisoning word* との距離は 10 km 以内の範囲となっている．*char_length* は *original word* と *poisoning word*

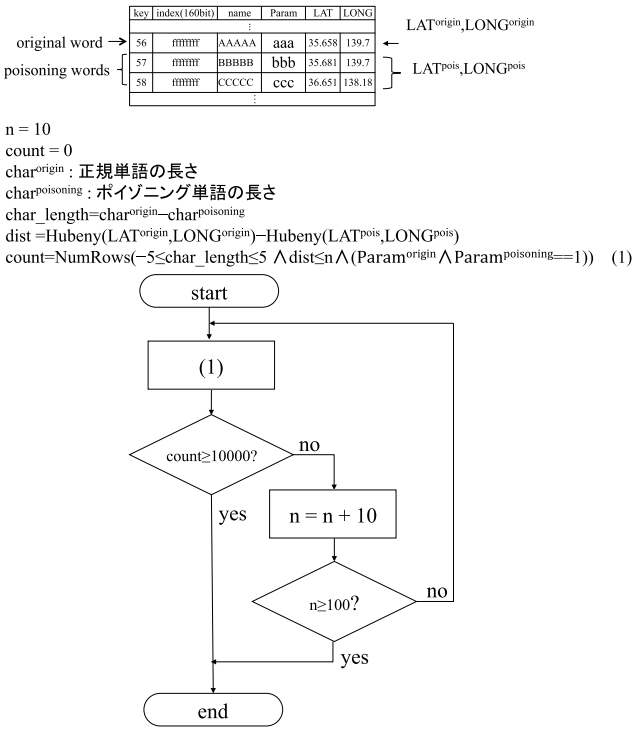


図 7 地域名，組織名に対するポイゾニング単語候補検索フロー
Fig. 7 Search flow of candidates for poisoning of names of regions and organizations.

との文字数差を表しており，式 (12) によって算出される．*dist* は *original word* に紐付けされた緯度と経度と，*poisoning word* に紐付けされた緯度と経度から算出した距離を表している．*dist* は式 (13) を用いて算出する．

$$dist = Hubeny(LAT_{origin},\ LONG_{origin}) - Hubeny(LAT_{pois},\ LONG_{pois}) \quad (13)$$

式 (13) の LAT_{origin} と $LONG_{origin}$ は *original word* に紐付けられた緯度，経度を表しており LAT_{pois} と LAT_{pois} は *poisoning word* に紐付けられた緯度と経度を表している．$Hubeny()$ は緯度と経度から距離を算出する関数である．

初回検索時に *count* が 10,000 通り以上になった場合はポイゾニング候補の検索を終了する．ポイゾニング候補の初回検索時に 10,000 通り未満であった場合は，n に 10 を加算し，*original word* と *poisoning word* との距離の範囲を拡張する．n が変更された後，テーブル内を *NumRows* 関数で再度検索を行う．ポイゾニング候補が 10,000 通りになるまで，*original word* と *poisoning word* との距離の範囲を拡張し検索を繰返す．*original word* と *poisoning word* との距離の範囲指定に用いている n の最大値は，100 とする．n が最大値になった時点で，ポイゾニング候補が 10,000 通り未満の場合であってもポイゾニング候補の検索を終了する．

ポイゾニング候補の中からポイゾニング採用単語を選出する手法は図 7 と同じフローである．以上により，県名，市名，町名，組織名の違和感の少ないポイゾニング採用単語の選出が可能である．

4. 実装

3 章で述べた手法の実装例として，渡部らが行った，AR によって紙文書の一部に情報を投影する技術を利用し，適切に情報をポイゾニングできるようにした．渡部らは，用紙に情報を印刷し，秘匿にしたい部分のみを空白にしている．スマートフォンなどのカメラと画面のついたデバイスをその用紙にかざすと，用紙の空白部分に文字が表示された状態で，画面に用紙が表示される．本論文の実装では，この空白部分に，3 章で述べた手法によりポイゾニングを行った情報を表示している．もちろん，レイアウトをともなう電子文書のポイゾニングに関しては，フォーマットが既知のものであればさらに容易である．AR の場合と同じく，画面上に表示されている文書の空白部分をポイゾニングするが，こちらはカメラによってリアルタイムに撮影されている用紙に書かれた文書ではないため，空白部分の位置が画面上で動かず固定されており，AR の場合と同様の手順で容易にポイゾニング可能である．いずれの場合でも，端末のロック解除に用いられる，4 桁の数字に基づいてポイゾニングが行われる．文書の作成に関しては，3.1 節冒頭で述べたように，文書の発行を行う制作者が，文書上で保護したい箇所を適切に選択して発行する．そして，同じく 3.1 節で述べられているとおり，そのタイミングで，秘匿にしたい部分には 2 倍のスペースが確保され，文書が成形される．ポイゾニング対象は，3.2 節で述べたポイゾニング手法に基づき，5 章で後述するように，データベースに登録されている語句から選ばれる．これは端末が表示を行う段階で決定される．

4.1 節にてマーカの生成，4.2 節にてサーバ，4.3 節にてクライアント，それぞれの実装方法を記述する．

4.1 マーカの生成

使用するマーカは渡部らの研究と同様に，QR コード [12] を ARtoolkit で使用できるように，仕様を合わせた形にする．書類識別 ID 以下 DocuID を QR コードに格納する．DocuID は，書類作成企業の識別を行う 8 バイトの CompanyCode，作成した書類の識別をする 6 バイトの paperSerial を連接させたものである．

4.2 サーバ

サーバのデータベースには，正文書用のテーブルと，ポイゾニング候補用のテーブルの 2 つが必要となる．正文書用テーブルの構造を**表 1** に示す．ポイゾニング候補用のテーブルには DocuID，閲覧許可者用の 4 桁のパスワード Pass，正文書用単語 T_word，正文書用単語のグルーピング区分情報 group を保存する．次に，ポイゾニング用テー

ブルを**表 2** に示す．ポイゾニング用テーブルにはポイゾニング候補採用に用いる index 値，index 値と，ユーザ専用乱数と入力パスワードのハッシュ値の一致率が同じになったときに優先順位を定めるための key を保存する．index のランダム値は，データベースにポイゾニング候補を追加するときに 160 bit のランダム値が割り当てられる．name にはそれぞれの名前や名称を保存する．

param は人名，商品名，病名によって適切に変更，増加させる．市名，町名，組織名用テーブルを**表 3** に示す．市名，町名，組織名用テーブルに保存される key，index，name の保存データは表 2 と同様の構造である．LAT には緯度，LONG 経度が保存される．

4.3 クライアント

クライアントプログラムは渡部らの研究を元に開発を行う．

クライアントのプログラムは QR コードの読み込みを行う．QR コードの読み込みをした時点で，パスワードの入力を求める．正規ユーザは閲覧者許可用の 4 桁のパスワードを入力する．パスワードの入力がされてから，クライアントプログラムは QR コードに格納された DocuID とパスワードをサーバに問い合わせる．DocuID とパスワードを受け取ったサーバは Pass に紐付けされた T_word を返す．

表 1 正文書用テーブルの構造
Table 1 Construction of table for original documents.

カラム名	容量 (byte)	カラム型
DocuID	13	CHAR
Pass	4	CHAR
T_word	MAX 65,535	TEXT
group	30	CHAR

表 2 人名，商品名，病名，県名用テーブルの構造
Table 2 Construction of table for names of people, products, diseases and prefectures.

カラム名	容量 (byte)	カラム型
key	4	INT
index	20	CHAR
name	MAX 65,535	TEXT
param	1	INT

表 3 市名，町名，組織名用テーブルの構造
Table 3 Construction of table for names of cities, towns and organizations.

カラム名	容量 (byte)	カラム型
key	4	CHAR
index	20	CHAR
name	MAX 65,535	TEXT
LAT	MAX 65,535	TEXT
LONG	MAX 65,535	TEXT

クライアントプログラムはサーバから T_word を受け取ると，AR 表示に切り替わり，文書上に T_word を AR 表示する．パスワードが違っていた場合サーバは，group の区分によって，3.2 節でのポイゾニング手法によりポイゾニングされた偽の情報を送信する．クライアントプログラムは偽情報を受け取ると，正しいパスワードの入力時と同様に AR 表示機能に切り替わり，ポイゾニングされた情報が AR 表示される．

5. 実験

本提案は，レイアウトを崩ずさず，違和感のない情報でポイゾニングを行う提案であるため，攻撃者がどのパスワードを入力したとしてもオリジナル情報が特定できないというシステムを実現したい．そのためには，実際に存在しうる名前や住所を用いても，適切にポイゾニングされている必要がある．

そこで，本提案手法により，オリジナル情報に対して違和感のないポイゾニングが行え，情報の確度が 0.01%になるか，つまりポイゾニングに使用する情報が 10,000 通り以上になるか実験を行った．その際の実験環境を**表 4** に示す．なお，今回のポイゾニングは，ユーザが文書を表示するタイミングでその候補が選出されている．このポイゾニング候補を選出する処理が，ユーザが文書を表示させる動作を行ってから，実際に文書が表示されるまでに掛かる時間に影響を与えるため，その処理の実行時間についても測定を行っている．考察については 6 章にて行う．

今回は名前および地名のポイゾニング手法の実験を行った．名前のポイゾニング手法は，オリジナル情報の文字数の範囲の上限と下限を変更しながらポイゾニングに使用する情報を抽出しているため，2 文字から 20 文字までの文字列をポイゾニングした場合のポイゾニングに使用される情報の総数とポイゾニング用の氏名の取得時間を測定した．名前用のポイゾニングテーブルには日本語名をローマ字にしたものが 56,907 件登録されている．テーブルの構成は表 2 のとおりである．

名前のポイゾニング手法の実験結果を**図 8** に示す．名前の結果から，名前の文字数が 2 文字である場合を除く文字列の長さであれば，ポイゾニングに使用する情報の総数が 10,000 件以上得られることが分かる．ポイゾニング情報の取得時間であるが，文字数が 6 文字以上からは上昇傾向にあることが分かる．5 文字以下の場合は取得時間にばらつきが見られる．

表 4 実験環境
Table 4 Experiment environment.

OS	Windows7 32 bit
CPU	Intel Core i7 3.2 GHz
Memory	4 Gbyte

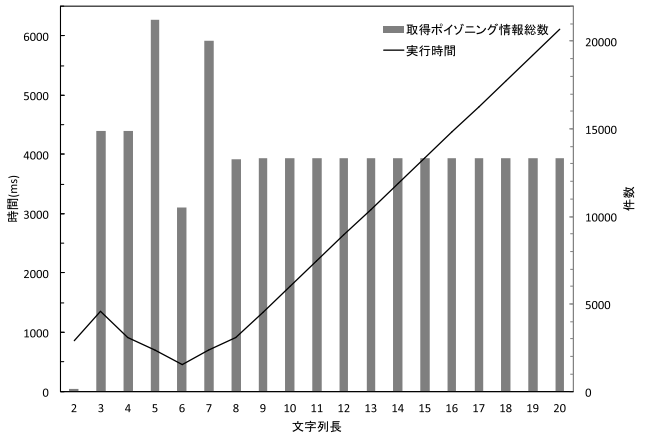


図 8 名前ポイゾニング実行結果
Fig. 8 Results of poisoning of names.

地名のポイゾニングでは，都道府県により市や町村区の密集度の違いや市の範囲が狭い広いなど環境が異なる．そのため，東京都と長野県の住所を実験対象とし，市町村の広さや密集度が異なっていても適切にポイゾニングされるのか実験を行った．オリジナル情報はデータベースに登録した東京都 23 区に属する町村名と長野県の各市に属する町村名からそれぞれランダム抽出した地名を使用する．実験としてポイゾニングに使用される情報の総数とポイゾニングされた偽情報の地名決定までの時間の計測を行った．もちろん，このポイゾニングは 3.2.3 項で述べたアルゴリズムに基づいており，式 (11) に示されているとおり，ポイゾニング候補の文字数も考慮されている．この実験に使用した東京都と長野県の町村名と，各町村の緯度経度のデータは国土交通省が提供している位置参照情報ダウンロードサービス [13] を使用した．テーブルの構成は表 3 のとおりである．

地域名のポイゾニングを実行し抽出された東京都 23 区に属する町村名のオリジナル情報とポイゾニング情報の一覧を**表 6** に示し，その際に掛かった実行時間とポイゾニング情報の総数を**図 10** に示す．同様に長野県の各市に属する町村名のオリジナル情報とポイゾニング情報の一覧を**表 7** に示し，その際に掛かった実行時間とポイゾニング情報の総数を**図 11** に示す．

地域名の結果から，町村名は数に限りがあるため，ポイゾニングに使用する情報の総数が 10,000 件を満たしていないことが分かる．長野県の一部町村名のポイゾニングに使用する情報の総数が 10 件に満たない結果が得られていることが分かった．

参考までに，AR の機能を用いて紙媒体に情報を追加表示する技術 [4] を利用して，デモ用に実装したものの写真を**図 9** に示す．なお，これはデモ用であるため，見やすいように関係のない情報をいっさい削除しているが，他に情報が書き込まれていてもシステムに影響はない．

また，オリジナル情報の文字数の 2 倍のスペースを確保し，そのスペースにオリジナル情報の文字数とは異なる文字数を表示し，見た者に違和感を与えるかどうかの実験を行った．結果を**表 5** に示す．

表の縦方向にはオリジナル情報の文字数が示され，表の横方向には表示された文字数のオリジナル情報の文字数

表 5 文字数の変化と違和感
Table 5 Relation between the number of words and the attention ratio.

文字数/差	−5	−4	−3	−2	−1	−0	+1	+2	+3	+4	+5
1	-	-	-	-	-	5	5	0	0	0	0
2	-	-	-	-	2	5	5	5	0	0	0
3	-	-	-	0	4	4	5	5	1	0	0
4	-	-	1	1	1	4	5	5	5	4	0
5	-	1	1	2	3	3	5	5	5	5	5
10	0	0	0	1	1	2	3	3	5	5	5

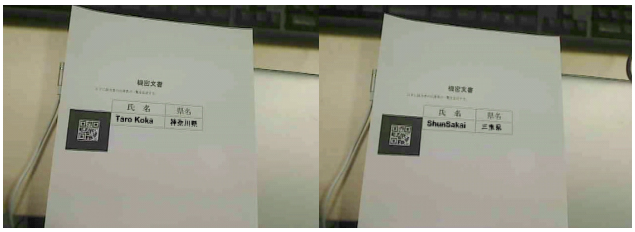


図 9 実装例
Fig. 9 Example of emplementation.

表 6 オリジナル情報とポイゾニング情報の一覧（東京都）
Table 6 Original words and words for poisoning (Tokyo Prefecture).

市区名	オリジナル情報	ポイソニング情報
足立区	西新井四丁目	保木間四丁目
荒川区	東尾久五丁目	東日暮里五丁目
板橋区	徳丸四丁目	成増一丁目
江戸川区	鹿骨二丁目	東松本一丁目
大田区	中馬込一丁目	東蒲田一丁目
葛飾区	立石六丁目	青戸五丁目
北区	中十条二丁目	赤羽西二丁目
江東区	三好四丁目	塩浜一丁目
品川区	東大井四丁目	西中延三丁目
渋谷区	千駄ヶ谷四丁目	東一丁目
新宿	四谷三丁目	西新宿二丁目
杉並区	高井戸東二丁目	今川四丁目
墨田区	業平二丁目	錦糸三丁目
世田谷区	松原三丁目	桜上水一丁目
台東区	浅草六丁目	池之端一丁目
千代田区	九段南二丁目	外神田一丁目
中央区	築地四丁目	日本橋本町三丁目
豊島区	長崎三丁目	駒込二丁目
中野区	弥生町二丁目	新井二丁目
練馬区	氷川台一丁目	向山三丁目
文京区	目白台三丁目	目白台一丁目
港区	麻布永坂町	新橋二丁目
目黒区	駒場二丁目	平町一丁目

との差が示されている．5 人の被験者に違和感について答えてもらい，違和感を覚えなかった人数を表に記した．なお，ここでは文字数に関しての違和感のみ答えてもらい，書かれている内容は考慮されないようにしている．結果として，最も違和感を覚えない文字数の前後 1〜2 文字に関しては，被験者のほぼ全員において違和感を覚えていないので，本提案のポイゾニング手法は，ポイゾニング候補がオリジナル情報とは文字数が多少異なる場合においても有効であるといえる．興味深いのは，文字数が増えるほど，最も違和感を覚えない文字数がオリジナル情報の文字数よりも若干増えた値であることである．この評価により，オリジナル情報の文字数の 2 倍のスペースを確保する場合に，文字数が増えるに従って確保するスペースを少しずつ小さくした方が，違和感を与えにくいということが分かった．

表 7 オリジナル情報とポイゾニング情報の一覧（長野県）
Table 7 Original words and words for poisoning (Nagano Prefecture).

市区名	オリジナル情報	ポイソニング情報
長野市	大字鶴賀	南千歳二丁目
松本市	笹部一丁目	波田
上田市	緑が丘三丁目	五加
岡谷市	長地出早一丁目	山下町二丁目
飯田市	東新町二丁目	宮の前
諏訪市	大字湖南	湖岸通り四丁目
須坂市	臥竜二丁目	大字坂田
小諸市	大字西原	乙
伊那市	美原	孤島
駒ヶ根市	赤穂	北町
中野市	新保	大字牛出
大町市	社	常盤
飯山市	大字木島	大字瑞穂豊
茅野市	中沖	豊平
塩尻市	広丘堅石	大字広丘高出
佐久市	根々井	香坂
千曲市	大字野高場	戸倉
東御市	県	新屋
安曇野市	穂高柏原	豊科南穂高
南佐久郡小海町	大字稲子	大字千代里
北佐久郡軽井沢町	大字追分	軽井沢
北佐久郡御代田町	大字広戸	大字御代田
諏訪郡富士見町	立沢	富士見
上伊那郡辰野町	大石平	大字澤底
上伊那郡箕輪町	大字福与	大字三日町
上伊那郡飯島町	田切	飯島
上伊那郡中川村	葛島	片桐
下伊那郡松川町	元大島	元大島
下伊那郡高森町	上市田	大島山
木曽郡上松町	駅前通り三丁目	栄町三丁目
木曽郡木曽町	新開	新聞福
北安曇郡池田町	大字陸郷	大字池田
北安曇郡白馬村	大字北城	大字神城
埴科郡坂城町	大字南条	大字南条
上高井郡小布施町	大字都住	大字山王島
上高井郡高山村	大字高井	大字高井
下高井郡山ノ内町	大字寒沢	平隠
下高井郡木島平村	大字往郷	上木島
下高井郡野沢温泉村	大字前坂	大字坪山
上水内郡信濃町	大字野尻	大字平岡
上水内郡飯綱町	大字豊野	大字坂口

6. 考察

5 章の結果の考察を記述する．名前のポイゾニング手法を行った際のオリジナル情報の確度を算出した．確度の算出は 1 をポイゾニングに使用する情報の総数を割った値とする．結果を**表 8** に示す．2 文字であった場合以外は確度が 0.01%以下になっている．2 文字で確度が極端に低くなってしまった要因は，ポイゾニング情報の文字列長が 4 文字と短かったために得られた総数が少なくなってしまったことである．文字数 9 文字以上の確度が同値になっている要因として，データベースに登録されている氏名の文字列が最大で 17 文字から 18 文字までの範囲しかないということが考えられる．さらに，図 8 から分かるとおり 9 文字以降は総数が同じであるにもかかわらず取得時間が上がってしまっている．ユーザがシステムの使用に耐えられる待ち時間は 3 秒以下 [14] であることから，文字数が 12 文字以上の場合は，ポイゾニング候補の検索範囲を一定にすることが望ましいと考えられる．

地域に関しても名前と同じように確度の算出を行った．結果を**表 9**，**表 10** に示す．確度が 0.01%以下になっていない理由は現存している地域名総数が有限であるためである．しかし，実在していない地域名を用いて確度を低くした場合，ユーザや攻撃者に違和感を与えてしまうためオリジナルの情報やパスワードの特定につながる恐れがある．そのため，地域名に関して確度は 0.01%でなくとも良いと考えられる．表 6，表 7 から，ポイゾニング用の地域名はオリジナル情報と地区に属した情報が抽出されていることから，違和感を与えることなくポイゾニングすることが可

表 8 名前ポイゾニングでの情報確度
Table 8 Accuracy of information with poisoning of names of people.

文字列長	確度（%）	文字列長	確度（%）
2	0.76336	12	0.00751
3	0.00673	13	0.00751
4	0.00673	14	0.00751
5	0.00471	15	0.00751
6	0.00952	16	0.00751
7	0.00499	17	0.00751
8	0.00753	18	0.00751
9	0.00751	19	0.00751
10	0.00751	20	0.00751
11	0.00751		

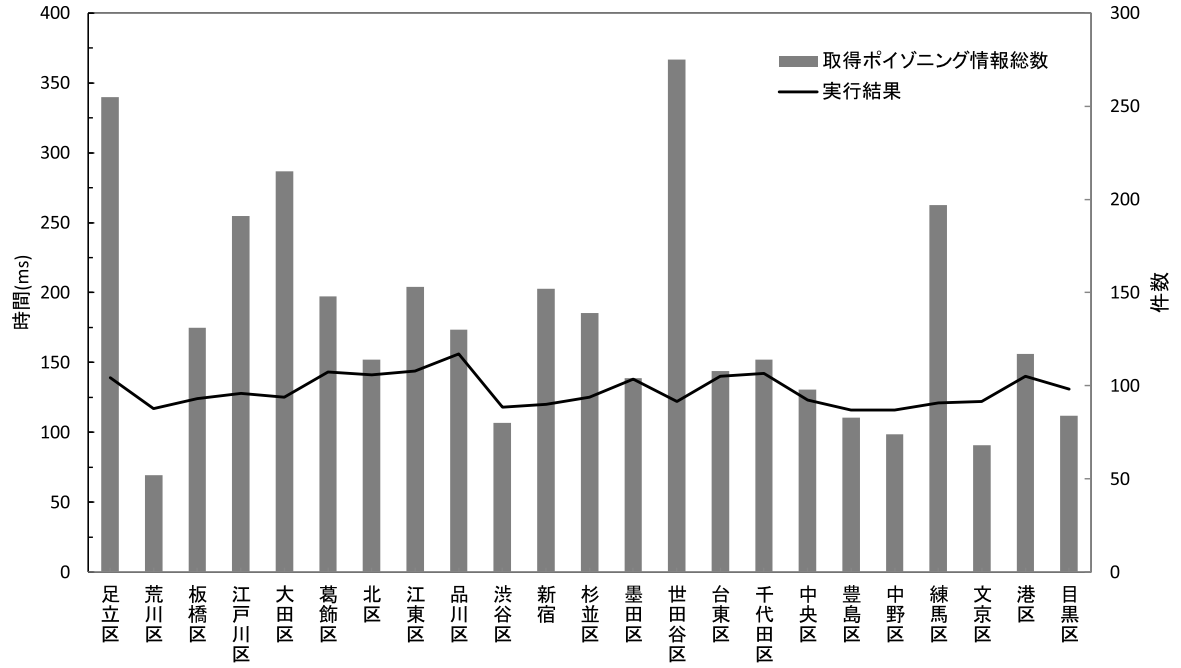


図 10 実行時間とポイゾニング情報の総数（東京都）

Fig. 10 Relation between execution time and the number of candidates for poisoning (Tokyo Prefecture).

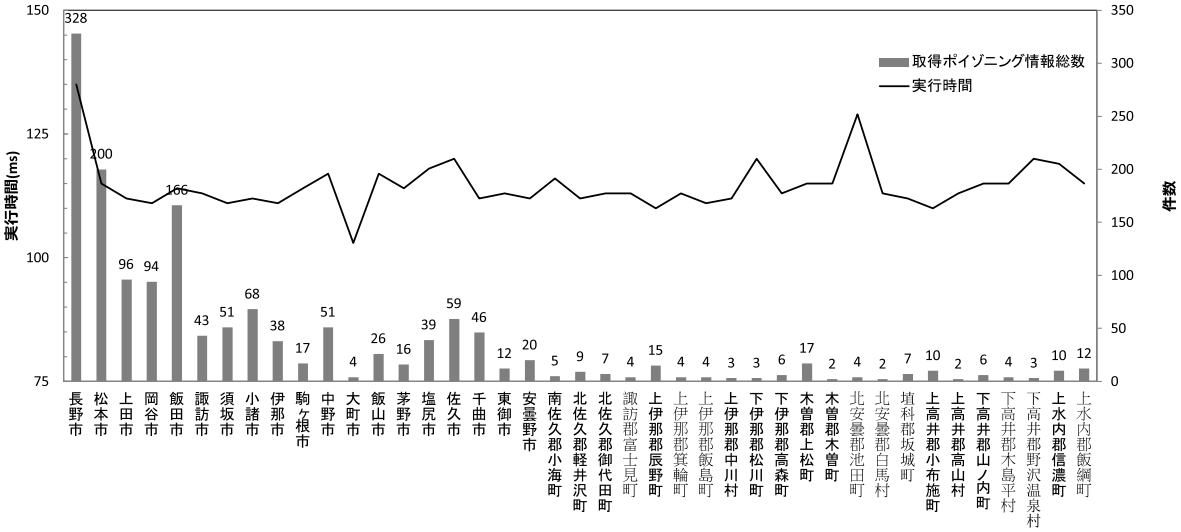


図 11 実行時間とポイゾニング情報の総数（長野県）

Fig. 11 Relation between execution time and the number of candidates for poisoning (Nagano Prefecture).

能であるといえる．

名前および地域名の確度が高くなる問題であるが，これはオリジナル情報 1 つに対する確度である．したがって，ポイゾニング対象になるオリジナル情報が複数の場合は確度が下がるため，問題なく運用することが可能であるといえる．

確度を論じる一方で，ポイゾニングされた文書の情報どうしに意味的な矛盾があれば安全性は下がるため，これについても考察する．まず，表 6，表 7，表 10 の地名に関して，ポイゾニング情報の中に実存しない地名はない．また，「丁目」などの下位の地名情報に対して，その上位の市区町村名などは必ず適切な組合せとなるため，地名の都道府県名，市区町村名，「丁目」などの組合せによる意味的な矛盾は生じない．

さらに，あるオリジナル情報に対して，どのポイゾニング候補が選出されるかについては，同じ語句であれば必ず同じポイゾニング候補が選出される仕組みであるため，この点においても，同一文書中で意味的な矛盾は生じない．たとえば，同一文書中の「八王子市」が，ある場所では「三鷹市」になり，別の場所では「町田市」になるようなこと

表 9 地域名ポイゾニングでの情報確度（東京都）
Table 9 Accuracy of information with poisoning of names of regions (Tokyo Prefecture).

市区名	確度（%）	市区名	確度（%）
足立区	0.3922	墨田区	0.9615
荒川区	1.9231	世田谷区	0.3636
板橋区	0.7634	台東区	0.9259
江戸川区	0.5236	千代田区	0.8772
大田区	0.4651	中央区	1.0204
葛飾区	0.6757	豊島区	1.2048
北区	0.8772	中野区	1.3514
江東区	0.6536	練馬区	0.5076
品川区	0.7692	文京区	1.4706
渋谷区	1.25	港区	0.8547
新宿	0.6579	目黒区	1.1905
杉並区	0.7194		

表 10 地域名ポイゾニングでの情報確度（長野県）
Table 10 Accuracy of information with poisoning of names of regions (Nagano Prefecture).

市区名	確度（%）	市区名	確度（%）
長野市	0.3049	北佐久郡御代田町	14.2857
松本市	0.5	諏訪郡富士見町	25
上田市	1.0417	上伊那郡辰野町	6.6667
岡谷市	1.0638	上伊那郡箕輪町	25
飯田市	0.6024	上伊那郡飯島町	25
諏訪市	2.3256	上伊那郡中川村	33.3333
須坂市	1.9608	下伊那郡松川町	33.3333
小諸市	1.4706	下伊那郡高森町	16.6667
伊那市	2.6316	木曽郡上松町	5.8824
駒ヶ根市	5.8824	木曽郡木曽町	50
中野市	1.9608	北安曇郡池田町	25
大町市	25	北安曇郡白馬村	50
飯山市	3.8462	埴科郡坂城町	14.2857
茅野市	6.25	上高井郡小布施町	10
塩尻市	2.5641	上高井郡高山村	50
佐久市	1.6949	下高井郡山ノ内町	16.6667
千曲市	2.1739	下高井郡木島平村	25
東御市	8.3333	下高井郡野沢温泉村	33.3333
安曇野市	5	上水内郡信濃町	10
南佐久郡小海町	20	上水内郡飯綱町	8.3333
北佐久郡軽井沢町	11.1111		

はない．

ただし，「東京都三鷹市にある東京工科大学」という文書を見た攻撃者が，東京工科大学が三鷹市にないことを知っていると，これがポイゾニングされた文書であることが見破られてしまう．

地名に限らず，攻撃者が，オリジナル情報の所有者に関係する知識を持っていればいるほど，ポイゾニングされた文書が見破られてしまう可能性が上がる．しかしながら，仮に不特定多数の間で，攻撃者がこの知識を共有しようとすれば，誰かが不正に情報にアクセスしていることが明らかになるため，攻撃者が知識を共有してオリジナル情報の候補を絞り込むことは現実的に困難であるといえる．

表 11 ポイゾニング情報のためのスペースに関する考察
Table 11 Consideration of space for poisoning.

確保されるスペース	選択確率
オリジナルの情報の長さの 2 倍 − 2 文字	10%の確率
オリジナルの情報の長さの 2 倍 − 1 文字	20%の確率
オリジナルの情報の長さの 2 倍	35%の確率
オリジナルの情報の長さの 2 倍 + 1 文字	20%の確率
オリジナルの情報の長さの 2 倍 + 2 文字	10%の確率

また，本提案手法で説明した，オリジナルの情報の長さの 2 倍のスペースが，ポイゾニングのためにつねに確保されている点について考察する．攻撃者が本提案手法の仕組みを知っている場合，オリジナルの情報の長さの 2 倍のスペースがつねに確保されていると，文字数がオリジナル情報とは異なるものが，ポイゾニング情報であると見破られてしまう．本論文の提案手法には取り入れられていないが，この問題を解決する手法として，確保するスペースを一定にしない方法を考案した．その手法を**表 11** に示す．

なお，どの長さのスペースが選択されるかの確率に関しては仮のものであり，実際にはポイゾニング候補の文字数の割合や，確保されているスペースと表示されている文字数についての違和感を考慮して，決定するのがよいと思われる．この手法を適用すれば，攻撃者にオリジナル情報の文字数を特定させないようにすることが可能である．

なお，0 となる数値の下位桁に関してのみは，攻撃者に知識がある場合にも，3.2.1 項で述べた手法により対応可能である．しかしながら，攻撃者が，適切な数値の上限や下限を知っていたり，数値が特定の値しかとらないことを知っていたりする場合には，本提案手法では対応できない．

本論文では，攻撃者が文書のポイゾニングされた箇所を見た際に，違和感なくポイゾニングが行えているかどうかのみの評価を行った．よって，攻撃者がそれ以上の前提知識を持つ場合には，安全性は本論文で述べた確度よりも低下する．攻撃者が前提知識を持つ条件は 2 つあり，1 つは文書の文脈から推測できるもの，もう 1 つは攻撃者がオリジナル情報の所有者に関して事前に知っている知識である．後者に関しては対応は困難であるが，前述したように不特定多数の間で，攻撃者がこの知識を共有することも困難である．前者に関しては，3.1 節の冒頭に記述したように，文書の発行を行う制作者がどの情報を保護するかに関して決定するため，どの情報をポイゾニングするのが有用かについてはこの制作者の判断に掛かっている．現時点では，制作者のポイゾニングを支援する手法はないが，文書の前後関係から攻撃者に知識を与えないようにポイゾニング箇所を選択できる手法を今後考えたい．

7. まとめ

ブロードバンドネットワークやスマートフォンの普及により，電子文書を扱う機会が多くなってきた．利便性の都合から，電子文書へのアクセスに複雑な認証方式を導入することは敬遠されており，情報流出が問題となってきている．AR 機能を用いて電子文書と紙文書を融合し，プライバシを保護する渡部らの手法や，暗号化された紙文書の一部に正規のデータをオーバレイする阿南らの手法も存在するが，パスワードが漏洩した場合の脅威については従来の電子文書と同様の問題をはらんでいる．電子文書に誤った情報を意図的に混入させる情報ポイゾニングの技術も注目されてはいるが渡部らの手法や阿南らの手法が対象とする紙面の文書においては，オリジナルの情報の文字数などのレイアウトが決まっており，不自然さをともなうことなく任意の候補で単純にポイゾニングを行うことは困難である．同様に，最近の電子文書においては見栄えの良い装飾となるようレイアウトが施されていることが多く，不自然さをともなわないポイゾニングを単純に行うことはできない．本論文では，ポイゾニングする項目が文書のレイアウトを不自然に変更しないように候補を絞り込む手法を提案した．同時に，意味的に不自然なポイゾニングにならないよう，候補をさらに絞り込んでいる．さらに，ポイゾニングによる情報の確度が一定値以下になるようにし，ポイゾニング効果を利用者に保証できるようにした．本手法は，情報漏洩を完全に防止しようとするものではなく，情報漏洩した際の被害を最小限に抑えるものである．よって，本手法は既存の情報漏洩防止技術と組合せて使用することが可能であり，電子文書がますます広まる社会にとって有用であると考えられる．

参考文献

[1] Lin, H., Ma, R., Guo, L., Zhang, P. and Chen, X.-J.: Conducting routing table poisoning attack in DHT networks, *International Conference on Communications, Circuits and Systems* (*ICCCAS*), pp.254–258 (2010).

[2] Ortega, A.P., Marcos, X.E., Chiang, L.D. and Abad, C.L.: Preventing ARP cache poisoning attacks: A proof of concept using OpenWrt, *Network Operations and Management Symposium*, pp.1–9 (2009).

[3] 阿南泰三，倉木健介，高橋 潤：紙の暗号化技術，FUJITSU, Vol.60, No.5, pp.490–495 (2009).

[4] 渡部友輔，手塚 伸，宮田宙和，宇田隆哉：AR を用いたアクセス制御可能な紙広告媒体の提案，暗号と情報セキュリティシンポジウム，4D1-2 (2011).

[5] Liang, J., Naoumov, N. and Ross, K.W.: The Index Poisoning Attack in P2P File Sharing Systems, *Proc. IEEE Infocom*, pp.1–12 (2006).

[6] Kong, J., Cai, W. and Wang, L.: *The Evaluation of Index Poisoning in BitTorrent*, Communication Software and Networks, pp.382–386 (2010).

[7] Lou, X. and Hwang, K.: Prevention of Index-Poisoning DDoS Attacks in Peer-to-Peer File-Sharing Networks, *IEEE Trans. Multimedia*, Special Issue on Content Storage and Delivery in P2P Networks (2006).

[8] 吉田雅裕，大坐畠智，中尾彰宏，川島幸之助：*Winny* ネットワークにおけるインデックスポイズニングの適用と評価，電子情報通信学会技術研究報告，Vol.108, No.203, NS2008–58, pp.93–98 (2008).

[9] 吉田雅裕，大坐畠智，中尾彰宏，川島幸之助：*Winny* ネットワークに対するインデックスポイズニングを用いたファイル流通制御方式，情報処理学会論文誌，Vol.50, No.9, pp.2008–2022 (2009).

[10] 吉田雅裕，大坐畠智，中尾彰宏，川島幸之助：P2P ファイル共有ネットワークにおけるインデックスポイズニングの動的な適用方式 (P2P)，電子情報通信学会技術研究報告，NS，ネットワークシステム，Vol.109, No.448, pp.279–284 (2010).

[11] 渡辺知恵美，荒井裕子，天笠俊之：ブルームフィルタを用いたプライバシ保護検索における攻撃モデルとデータ撹乱法の一検討，日本データベース学会論文誌，Vol.8, No.1, pp.113–118 (2009).

[12] 株式会社デンソーウェーブ：QRcode.com（オンライン），入手先 ⟨http://www.qrcode.com/⟩（参照 2012-05-13）．

[13] 国土交通省：位置参照情報ダウンロードサービス（オンライン），入手先 ⟨http://nlftp.mlit.go.jp/isj/⟩（参照 2012-11-27）．

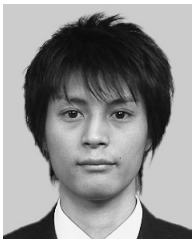
[14] Forrester Consulting on behalf of Akamai Technologies, Inc.: eCommerce WebSite Performance Today (online), available from ⟨http://www.damcogroup.com/white-papers/ecommerce_website_perf_wp.pdf⟩ (accessed 20012-11-14).

石井 順也

1988 年生．2011 年東京工科大学コンピュータサイエンス学部コンピュータサイエンス学科卒業．2013 年同大学大学院博士前期課程修了．現在，TIS 株式会社勤務．

ノールアフィザ マットラザリ

2002 年茨城大学情報工学科卒業．NTT MSC Malaysia，HP MSC Malaysia に勤務．2013 年東京工科大学大学院バイオ・情報メディア研究科コンピュータサイエンス専攻博士前期課程修了．現在，東京工科大学大学院博士後期課程に在学中．情報セキュリティの研究に従事．



手塚 伸 （学生会員）

2006 年東京工科大学情報工学科卒業，2008 年同大学院バイオ・情報メディア研究科前期博士課程修了．現在，慶應義塾大学大学院理工学研究科開放環境科学専攻後期博士課程に在学中，慶應義塾大学 ITC 本部助教．ネットワークセキュリティの研究に従事．

宇田 隆哉 （正会員）

1975 年生．1998 年慶應義塾大学理工学部計測工学科卒業．2000 年同大学大学院理工学研究科計測工学専攻前期博士課程修了．2002 年同大学院理工学研究科開放環境科学専攻後期博士課程修了．博士（工学）．現在，東京工科大学コンピュータサイエンス学部講師．ネットワークセキュリティの研究に従事．2002 年 IFIP/SEC 2002 Best Student Paper Award 受賞．電子情報通信学会会員．

木下 俊之 （正会員）

1977 年東京大学大学院理学系研究科数学専攻修士課程修了．同年（株）日立製作所入社．同社システム開発研究所にて，計算機性能評価，オペレーティングシステム，インターネットセキュリティの研究に従事．2005 年東京工科大学コンピュータサイエンス学部教授．博士（理学）．技術士（情報処理部門）．電子情報通信学会，ACM 各会員．