Vol.2014-SPT-8 No.35
2014/3/28

暗号技術の信頼情報を提供するクライアント側機構とユーザインタフェースの提案

岩橋 敏幸[1]　金岡 晃[1]

概要：クラウドコンピューティングの発展やビッグデータの活用が進む中で、プライバシ保護技術にその注目が集まっている。プライバシ保護はさまざまな手法が提案されており、とくに情報の秘匿性を保持したまま計算が可能な手法はその注目が高い。一方で、一般の利用者から見ると情報が秘匿されているままでありながら計算が可能な手法は直感的な理解が困難であり、プライバシ保護技術そのものについての疑問が根深く存在する。本稿ではこれらの疑問を解消するためのクライアント側機構とユーザインタフェースの提案を行う。

1. はじめに

暗号技術は電子商取引による通信の秘匿化や認証など既に広く使われているほか、学術研究としても様々な暗号技術が今もなお多く研究が行われている。これらの暗号技術は現実利用となると一般ユーザが意識することなく利用できることが多いが、クラウドコンピューティングの時代を迎え情報保管や処理の外部委託が盛んになると、委託先やあるいはその通信路上で適切に暗号技術が用いられているかユーザ自身が確認可能になることが重要になる。

しかし、クラウドコンピューティングは技術的に深まっているものの、ユーザ側から見たときのその技術への信頼性確立がされているとはいい難い。クラウドコンピューティングが発展すると同時に、プライバシを技術的に保護する技術開発も活発になってきた。データを秘匿したまま計算をする秘匿計算手法や、データからプライバシが漏えいすることを防ぐ匿名化技術など、様々なアプローチが取られている。プライバシ保護技術についても技術の進展に目を見張るものがあるが、ユーザ側から見た場合の信頼性確立はプライバシ保護技術についてもまだアプローチがされていない。

ユーザ側からみた場合の信頼性確立についてはまだ深い研究がされていないが、技術的に類似あるいは応用が可能な研究がいくつかされている。暗号化を確認する研究として Dijk ら [1] の研究はクラウドに保存されているファイルが本当に暗号化されているかを確認する方法を提案した。Dijk らの手法ではファイルの暗号化状態を確認できる保存方式でクラウド上で保存し、チャレンジ・レスポンス方式を用いて安全に確認していた。また、Grobert ら [2] の研究はバイナリから利用されている暗号技術を探る手法を提案した。Grobert らの手法ではバイナリに存在する暗号技術ごとの特徴を収集し、その収集した情報と照らし合わしていた。先述の通り、これらの研究は暗号化を確認する手段の研究であってユーザに暗号技術の信頼情報を伝え保障するものではないが、応用として利用できる可能性はある。

本論文では、利用されている暗号技術が信頼のできるものであるかどうかをユーザに示すことを目的として、ユーザ側で信頼情報を提示するためのモジュールとそのユーザインタフェースを提案する。本論文では、以下の点について整理と提案を行った。

- 暗号技術の信頼情報を提供するための条件
- 暗号技術の信頼情報を提供するためのフレームワーク
- 信頼情報として提供される情報の検討
- 信頼情報を提供するインターフェース

まず暗号技術の信頼情報を提供するための条件ではどういった要素が信頼情報を提供するべきかを考え、整理した。次に暗号技術の信頼情報を提供するためのフレームワークではまとめた要件から信頼情報をユーザ側で提供するための仕組みと信頼の判断方法について考察している。信頼情報として提供される情報の検討ではどのような情報を信頼情報として提供するべきかを大きく 2 種類に分けて考察した。信頼情報を提供するインターフェースでは信頼情報を提供するインターフェースとして適用可能な手法をまとめた。

暗号技術として検索可能暗号を対象に定め、信頼情報を提示するためのインタフェースとしてウェブブラウザによ

[1] 東邦大学
2-2-1, Miyama, Funabashi, Chiba 274–8510, Japan

る表示画面の実装を行った。実装した表示画面を評価するためにユーザ実験にてユーザに実装した表示画面による信頼情報の提供を行った。その結果、ユーザ実験より表示画面を見た半数以上の人間に信頼を与えることができたかどうかについて良い評価を得ることができたため、第 3 者に対する信頼を与えることに寄与していることがわかった。そして、提案手法はユーザに「第 3 者への信頼情報をユーザに提供する」という影響を与えるだけではなく「預けている情報のプライバシへの注目」と「第 3 者へデータを預けることへの不安を明らかにする」という 3 点の影響を与える可能性も示された。

2. 関連研究

データが実際に暗号化されているを確認する手法に関しては、いくつかの研究がされている。Dijk ら [1] はクラウドコンピューティング環境に保存されているファイルが本当に暗号化されているかを確認する方法を提案した。Dijk らの手法ではファイルの暗号化状態を確認できる保存方式でクラウド上で保存し、チャレンジ・レスポンス方式を用いて確認を行う。その他、

Grobert ら [2] の研究はバイナリから利用されている暗号技術を探る手法を提案した。Grobert らの手法ではバイナリに存在する暗号技術ごとの特徴を収集し、その収集した情報と照らし合わしていた。以上 2 つの研究は暗号化を確認するという点では共通点があるが、対象となっている暗号技術が特に検索可能暗号を対象にしたのものではない。

有効な警告表示の手法についての研究として Sunshine ら [3] の研究はウェブブラウザの SSL 通信の警告についてどういう表示が効果的かを調べる研究といったものであったが、FireFox の警告表示がリスクの高い銀行のサイトで表示された場合、リスクの低い図書館のサイトで表示された場合の両方で警告を無視される確率が低かったため効果的な表示画面であるといった結論であった。

また、Akhawe ら [4] の研究はウェブ ブラウザの警告表示でどのような表示が効果的かを調べる研究という内容であったがこちらの研究ではマルウェアの警告、SSL 通信の警告の二種類で検証しておりどちらの場合でも FireFox の警告表示が効果的であったという結論であった。その結論に至った理由としてどちらの警告も無視される確率が低かった事が挙げられていた。またマルウェアの警告表示では表示を無視するのに必要なクリック回数が一回であるのに関わらず必要なクリック回数が多い他のブラウザに比べて無視される確率が低かったために合理的な表示画面であると考えられると同時に表示を無視するためのリンクが右下の位置にあるため、意図して見落とすような配置にしてある可能性を考察していた。

これらの研究対象の表示画面は表示する内容が SSL 通信の警告、またはマルウェアの警告となっているため本論文における研究対象の表示画面とは表示する内容、目的がが異なる。

3. 提案するフレームワーク

本論文では利用されている暗号技術が信頼のできるものであるかどうかをユーザに示すことを目的として、ユーザ側で信頼情報を提示するためのモジュールとそのインターフェースを提案する。

3.1 暗号技術の信頼情報を提供するための条件

暗号技術の信頼情報を提供するには、ユーザが信頼している人による提供が必要であり、ユーザが信頼している人は必ずしも暗号技術の利用者（クライアントアプリ作成者、サーバアプリ作成者、サーバアプリ運用管理者）ではない。むしろ暗号技術の利用者以外で使われることをここでは考える。そうすると、要件としては以下のことになる。

- 暗号技術の実施過程の内容を確認することはできず、暗号技術そのものはブラックボックスとして考える必要がある
- ブラックボックスとして利用するために、信頼情報の提供するためにはその対象となる暗号技術への入力と出力だけから判断することが求められる

3.2 暗号技術の信頼情報を提供するためのフレームワーク

3.1 節の要件を考えるとユーザ側での暗号利用アプリの入出力を観測する必要があるので、ユーザ側（クライアント、サーバ側いずれにしても）で観測する仕組みが必要である。よって図 1 のように観測ポイントでその入出力を観測し、その信頼を判断する。暗号技術によっては、判断す

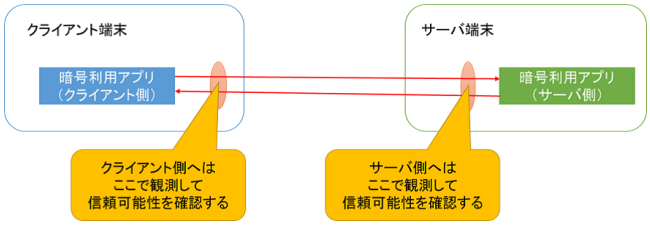

図 1 信頼の判断方法

るにあたり信頼確認をしたい時点での入出力だけでは判断できない。なので図 2 のように外部に関連した知識を保管した仕組みを持たせることで、入出力情報に加え、外部の知識を利用して信頼確認を行う。

3.3 信頼情報として提供される情報の検討

信頼情報として提供される情報は大きく 2 種類に分けることができる。1 つはすべての暗号技術に共通した方式であり、もう 1 つは暗号技術に依存した確認方法である。本論文ではでは共通した方式としてコンフォーマンステスト

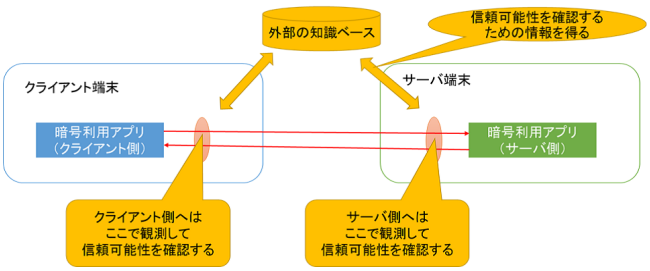

図 2 外部の知識を利用した信頼確認方法

と乱数性確認を提案する。

(1) コンフォーマンステスト

通信を介して複数のプレーヤーがデータの処理を行う場合、その通信方法やデータの中身は標準的な仕様として定められている。たとえば SSL/TLS の場合では TLS1.0 は RFC 2246、TLS1.1 は RFC 4346、TLS1.2 は RFC 5246 で定められている。暗号の場合、RSA 暗号の鍵は PKCS# 1 として定められている。各アプリケーションはどういった暗号を使うかが確認できると仮定すると、入出力から各仕様のデータフォーマットや通信方法を確認することでその暗号が利用されているかどうかを形式上で確認できる。このテストだけでは暗号技術への信頼形成への影響は大きくないことが予想されるが、最低限のレベルとしての保証は提供可能である。

(2) 乱数性の確認

さまざまな暗号技術が提供される場合、暗号技術により保護された情報は高い乱数性を持つことが期待されている。そこで入出力のデータの乱数性を確認することで、少なくともその暗号技術に対して入力されたデータあるいは出力されたデータが、アプリケーションが主張する暗号技術を実装しているかどうかをある程度保証することが可能である。乱数性の確認はさまざまな手法がある。たとえば NIST による乱数性の検定などが代表的である。NIST による乱数検定は以下の 16 種類の検定法がある。

(a) 頻度検定 (frequency test)
(b) ブロック単位の頻度検定 (frequency test within a block)
(c) 連の検定 (runs test)
(d) ブロック単位の最長連検定 (test for longest run of ones in a block)
(e) 2 値行列ランク検定 (binary matrix rank test)
(f) 離散フーリエ変換検定 (discrete fourier transform (spectral) test)
(g) 重なりのないテンプレート適合検定 (non-overlapping template matching test)
(h) 重なりのあるテンプレート適合検定 (overlapping template matching test)
(i) Maurer のユニバーサル統計検定 (Maurer's universal statistical test)
(j) Lempel-Ziv 圧縮検定 (Lempel-Ziv compression test)
(k) 線形複雑度検定 (linear complexity test)
(l) 系列検定 (serial test)
(m)近似エントロピー検定 (approximate entropy test)
(n) 累積和検定 (cumulative sums (cusums) test)
(o) ランダム偏差検定 (random excursions test)
(p) 種々のランダム偏差検定 (random excursions variant test)

NIST の検定では、各検定ごとに $p-value$ が得られる。$p-value$ とは、検定で出力される統計量の正規分布もしくは、カイ 2 乗分布において、それよりも偏った統計量が発生する確率を表したものである。$p-value < 0.01$ の時に良い乱数ではないと判断する。各検定では、複数の標本系列 (NIST では 1000 程度を推奨) に対し検定を行い、

(a) $p-value$ の一様性
(b) $p-value$ が 0.01 より大きくなる場合

から乱数列の評価を行う。(a) の場合には得られた $p-value$ が区間 [0.1) で一様に分布しているかどうかを調べるために、[0.1) を 10 の区間に分割し、分割した区間ごとの頻度が一様になっているかどうかをカイ 2 乗検定により得られた $p-value$ が 0.0001 以上ならば、乱数列は良い乱数であると判断する。(b) の場合には標本の数を m とした時、0.01 以上となる $p-value$ の数の割合が $0.99 \pm 3\sqrt{\frac{0.99\times0.01}{m}}$ の範囲に入っている場合は、乱数列は良い乱数であると判断する。

(3) 暗号技術に依存した確認方法

暗号技術はさまざまな手法があるため、それに従った方法での確認をすることが可能である。技術としては、暗号技術に対する攻撃手法として研究されているサイドチャネル攻撃の手法が適用可能である。サイドチャネル攻撃では暗号の処理時間が処理するデータに依存する点を利用して鍵を推定する処理時間解析や IC チップの消費電力を解析することで処理されている秘密情報を取り出す電力解析などがある。

処理時間解析は、暗号が処理する時間を計測することでその動作内容を知り、暗号に用いられた鍵などを推定する技術であり、攻撃手法として研究が行われている。また近年の暗号は処理時間解析による攻撃に耐性があるように設計・改良・実装が行われているが、攻撃まで至らずにこの技術を応用することで本研究で仮定されているブラックボックスの環境で利用されている暗号技術の特徴を確認することが可能であると考えられる。

3.4 信頼情報を提供するインターフェース

信頼情報を提供するインターフェースはさまざまな手法が適用可能である。

- クライアント側がブラウザアプリの場合：ブラウザの拡張機能として実現
  - ダイアログを出す
  - コンテンツを追加・修正する

  これらはいずれもクライアント側で実施する JavaScript により第 3 者の提供として実現可能である。
- 独立したアプリの場合：OS の機能あるいは他のアプリとして実現
  - 通知を出す : Windows の場合だとタスクトレイ、Android や iOS といったスマートフォン・タブレットの場合では OS が持つ通知機能の利用
  - 信頼情報提供アプリを独立して提供：サーバ側の場合、監視ソフトの 1 つとして。

4. フレームワークのシステム試作と評価実験

4.1 実験における目的と実験概要

実際にインターフェースを用意して、ユーザ実験を行うことで提案手法の有用性を評価することを目的にユーザ実験を行った。実験の概要は提案画面を導入したシステムと導入していないシステムの 2 種類を用意し、それぞれでユーザに作業を行ってもらい、作業終了後に事後アンケートを実施してユーザに提案画面について聞くというものである。また、実験で利用する暗号技術として Curtmola らによる検索可能対称暗号である SSE-1 を選択した [5]。

4.2 実験における想定環境

実験における想定環境として、ユーザは第 3 者のサービスを利用して日記をつけており、このサービスはブラウザによって閲覧するものであり、検索可能暗号を用いて検索可能というシステムとした。このシステムで用いられている検索可能暗号は Curtmola らによる SSE-1 であり、提案手法の適用ポイントはブラウザでのダイアログ表示とした。

暗号化されているかを確認するシステムは検索可能暗号を利用したユーザとサーバ系のやり取りを観測し、SSE-1 の機能の 1 つである Trapdoor の実施後に通信で送られるデータを取得しその内容を調べることで暗号化されているか否かを分析し、分析結果を CGI を利用して渡す。分析内容は以下である。

- コンフォーマンステスト
  使用されている暗号方式にそったフォーマットでデータが保存されているかを分析する。
- 乱数性の確認
  保存されているデータはどの程度の乱数性のもと保存されているかを NIST の乱数性検定ツールを使って分析する。結果をスコア化して表示する。
- 暗号技術に依存した確認
  まず暗号方式特有の分析が必要かを判断し、必要の場合には分析を行う。

また、乱数性の確認と暗号技術に依存した確認はユーザが検索可能暗号を利用した瞬間に流れているデータだけではデータが少なくその確認が確実には取れないために以下の 2 つの方法でで分析する方式にした。

- 利用する瞬間のデータの分析
  利用した瞬間のデータを上記の 3 つの分析内容でチェックする。
- 各暗号方式におけるこれまでの分析結果をまとめた外部の知識ベースとの照合
  利用している暗号方式の過去の分析結果から信頼できるかをチェックする。

今回対象とした SSE-1 では単なる 2 つの AES の暗号データが得られるため方式特有のチェックは行っておらず、乱数性の保証するためには乱数を長く評価しないとならないことから以下のように外部の知識ベースをあらかじめ構築した。

(1) SSE-1 の Trapdoor 対象として検索キーワードを 514054 個用意
(2) 514054 個のキーワードより得られた Trapdoor のバイナリデータを 1 つのデータとして統合
(3) 統合したデータを NIST の乱数性検定ツールを適応し、乱数性を確認
(4) 得られたスコア情報を外部の知識ベースとして保存

4.3 ユーザテストの環境

実験でユーザに使用してもあったブラウザは Internet Explorer 10 であり、利用してもらった実験選択ページや実験の導入ページ、日記の検索ページ、検索結果のページ、作業後のページ、事後アンケートのページはすべて Perl を用いた Web アプリケーションとして実現した。実験の大まかな流れを以下に示す。

(1) 実験の導入ページに置いて、実験内容を読んでもらい作業開始
(2) 作業は日記のキーワード検索と検索結果から気になった日記の内容をメモ欄に転記するもの
(3) 作業終了後は終了のメッセージを表示し、事後アンケートに移
(4) 事後アンケートに回答後実験終了

実験は 2 種行い、1 つは提案するシステムを導入していないシステムで行い、もう 1 つは提案システムを用いて行った。

実験の導入ページでは被験者の作業時に提案画面に対して意識を持たないように実験の目的を「ネットワークからのさまざま攻撃を検知するシステムが導入されている PC で作業してもらってシステムのさらなる軽量化が必要か

を事後アンケートで調査する」と被験者に伝え作業してもらった。作業終了後の事後アンケート実施前に本当の実験の目的である「非常にプライベートな情報を第 3 者に預けているときのデータ保護に信頼感を感じているか」を被験者に伝えた。

提案手法の適用ポイントとなるブラウザでのダイアログ表示のための表示画面は jQuery UI を用いて作成した。表示画面をユーザにしっかりと見てもらうため、表示する際には表示画面以外が暗転し、表示画面を閉じるまで他のウェブページの要素は有効にならなくなるようにした (図 4)。表示画面のメイン部分（図 5）の情報を見やすく簡潔にするためにチェック結果の詳しい情報は表示画面内のリンクで小窓のウェブページで開いて確認できる仕様（図 6）にした。この表示画面が提案画面となる。

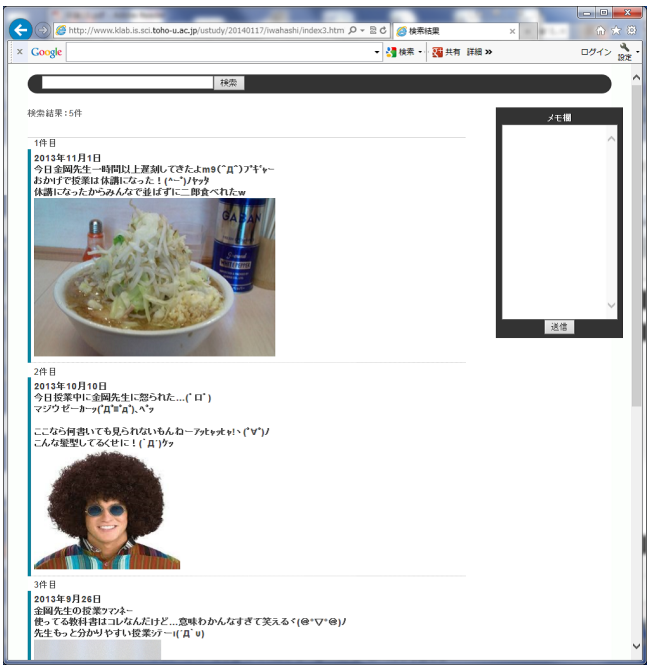

図 3 検索結果のページの一例

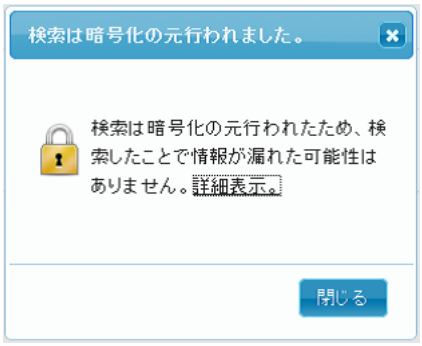

図 4 作成した表示画面－通知ダイアログ

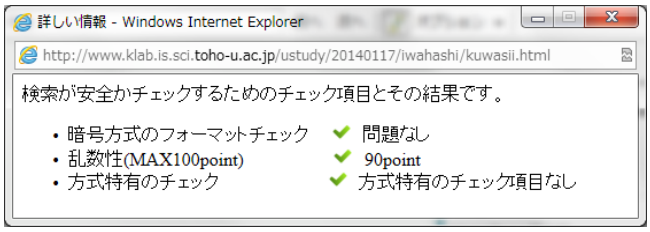

図 5 作成した表示画面－詳細結果ダイアログ

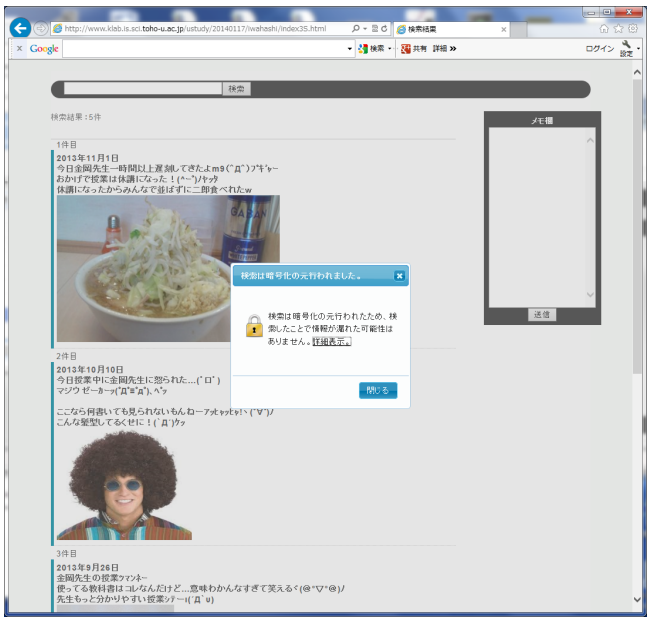

図 6 作成した表示画面－実際の見え方

4.4 ユーザテストの実施概要と詳細

[実験スタイル] ロールプレイ型

[実験対象] 東邦大学理学部情報科学科に所属する学生

[ロールプレイ（役割演技）の内容]

- 19 歳、男性、大学生 1 年生、交際している女性あり。大手コンビニエンスストアでアルバイトをしている。
- インターネット上のサービスを使い日記をつけることが日課。だがブログと違い誰にも公開はしていない。
- ある日、あずけてある日記を読み返したくなったが、いつ書いた日記なのか忘れたためサービス上でキーワード検索することで見つけることにした。（注：日記のサービスでは、日記データはサービスの提供事業者が持つインターネット上のサーバに保管されているが、暗号化したまま検索可能なようになっている。）
- キーワードを検索し、読み返すこととした。

[作業内容]

(1) 指定した 5 つのキーワードを順に検索してもらう。
(2) 1 つのキーワードを検索するごとにその検索結果を全て読み 1 番印象に残った日記の内容を画面右にあるメモ欄に記載する (コピー&ペーストでかまいません)。
(3) 記載を終えたらメモ欄下部にある送信ボタンを押して

メモの内容を送信する。

(4) 5 つのキーワードすべてを検索し、それぞれメモを送信したら作業終了。

[事後アンケートの内容]

[設問 1] 年齢と性別をお答えください。(自由記述)

[設問 2] 調べた内容は機微(プライベート)な内容でしたか?(5 段階の選択肢より選択)

[設問 3] 第 3 者にデータを預けていることに不安を感じましたか?(5 段階の選択肢より選択)

[設問 4] ロールプレイにおいて実際に検索作業をしていただきましたが、検索とデータの保管に利用した第 3 者は信頼できましたか?(5 段階の選択肢より選択)

[設問 5] 設問 4 の回答を選んだ理由をお答えください(自由記述)

[設問 6] 検索時に出てきた暗号化してあるというダイアログ(表示)はあなたに信頼感を与えましたか?(提案画面を導入しているシステム利用者のみ回答。5 段階の選択肢より選択)

[実験のねらい]

ユーザには日記を検索してもらうがその検索結果で表示される日記の内容ががとても機微な内容となっているため検索することで日記の内容が漏洩してしまわないか不安になる。提案画面を導入しているシステムでは検索結果を表示するタイミングで図 6 に示したように提案画面が表示されることでユーザに信頼を与えることをねらいとしている。提案画面はキーワードを検索し検索結果が表示される毎に表示される。

4.5 ユーザテストの評価

ユーザテストでの被験者は合計で 22 人であった。22 人のうち男性は 11 名、女性は 11 名であった。また被験者の年齢は 18〜22 であった。11 人(男性 7 人、女性 4 人)が提案画面を導入していない実験 1、11 人(男性 4 人、女性 7 人)が提案画面を導入した実験 2 を行った。

実験 1 では、日記の内容を「非常に機微である」と答えた被験者は 4 名(36.4%)、「機微である」と答えた被験者は 2 名(18.2%)であった。一方、実験 2 では「非常に機微である」と答えた被験者は 5 名(45.5%)、「機微である」と答えた被験者は 4 名(36.4%)であった。日記の内容を「非常に機微である」または「機微である」と感じた被験者の割合は、実験 1 の被験者が 6 名(54.6%)なのに対し、実験 2 の被験者では 9 名(81.9%)と多い割合を示しており、提案画面の提示により日記の内容をあらためて機微であると感じた様子がうかがえる。ここから、提案画面が日記内容のプライバシにあらためて注目をさせる機会を与えている可能性が考えられる。

図 7、8 は設問 3 の回答をまとめたものである。実験 1 における被験者の回答では、第 3 者にデータを預けることへの不安を「強く感じた」が 3 名(27.3%)、「感じた」が 3 名(27.3%)となっている。実験 2 における被験者の回答では、第 3 者にデータを預けることへの不安を「強く感じた」が 5 名(45.5%)、「感じた」が 5 名(45.5%)となっている。「強く感じた」あるいは「感じた」と回答した被験者数が実験 1 では 6 名(54.6%)なのに対し、実験 2 では 10 名(91%)と多くなっていることがわかる。また実験 2 では「感じない」あるいは「全く感じない」と回答した被験者数が 0 名だったことも特徴的である。これらの結果から、提案手法により第 3 者へデータを預けることに対する不安が増大することが示唆され、提案手法は第 3 者へ預けられたデータがプライベートであるかを自覚させると同時にデータを預けることへの不安を持たせている可能性があることを示している。

図 9,10 は設問 4 の回答をまとめたものである。実験 1 における被験者の回答では、第 3 者への信頼について「強く信頼できた」が 0 名、「信頼できた」が 1 名(9.1%)となっている。実験 2 における被験者の回答では、第 3 者への信頼について「強く信頼できた」が 1 名(9.1%)、「信頼できた」が 1 名(9.1%)となっている。「考えなかった」が実験 1 で 6 名(54.5%)、実験 2 で 7 名(63.6%)と多数を占めていることから、第 3 者への信頼については考慮されていないことがうかがえる。実験 2 で「強く信頼できた」と回答した被験者が 1 名おり、提案手法により第 3 者サービスに対する信頼形成を促した、と考えることも可能であるが、回答数が 1 名であることから、強くそれを支持するものとはいいがたく、今後のより多くの被験者による実験によりこの点は明らかになると考えられる。

設問 5 を回答は 19 件であった。設問 4 において信頼について「考えていない」という回答が多かったために、信頼について考慮されていない旨が書かれた回答が多い傾向にあるが、その中にも「自己責任であると思った」「今ではみんながこのような書き込みを当たり前にしている」といった、リスクについて自覚しながらも考えないという姿勢をもった被験者もおり、興味深い。

設問 6 は実験 2 を行った被験者のみに行った設問である。図 11 は設問 6 の回答をまとめたものである。ここで、提案手法が信頼を与えたかどうかについて「強く与えた」と回答した被験者が 3 名(27.3%)、「与えた」と回答した被験者が 3 名(27.3%)と、合わせて 6 名(54.6%)良い評価を与えている。一方で「与えていなかった」が 1 名(9.1%)、「全く与えていなかった」が 2 名(18.2%)と、合わせて 3 名(27.3%)であり、提案手法が第 3 者に対する信頼を与えることに寄与していることがわかる。一方で、これらの信頼への寄与の効果があるとすると、設問 4 において実験 1 と実験 2 において差が発生することが期待されるが、先述の通り、回答者数が少ないためにその効果を強く

示すものとは言えないものとなっていることに注意が必要である。

これらの結果から、提案手法が被験者にあたえている影響については、以下の可能性があることが示されたと言える。

- 預けている情報のプライバシへ注目を持たせる
- 第 3 者へデータを預けることへの不安を明らかにする
- 第 3 者への信頼情報をユーザに提供する

3 点目は本研究で主眼とされていたものであったが、先述の通り、被験者数の少なさから効果があると強く断定することは難しいが、その可能性を否定するものではなかった。今後さらにユーザ実験を進めてその評価を確実にすることが求められよう。一方で、提案手法により預けているデータが機微であるかどうか、また預けることに対する不安の明確化、という 2 点が示されており、通信が発展し意識せずに通信を行っている現代において、意識しない通信による無意識の信頼の醸成に対してあらためて信頼の確立の重要性を認識させる要因としての意味も持つことが示唆されており、興味深い結果と言えよう。

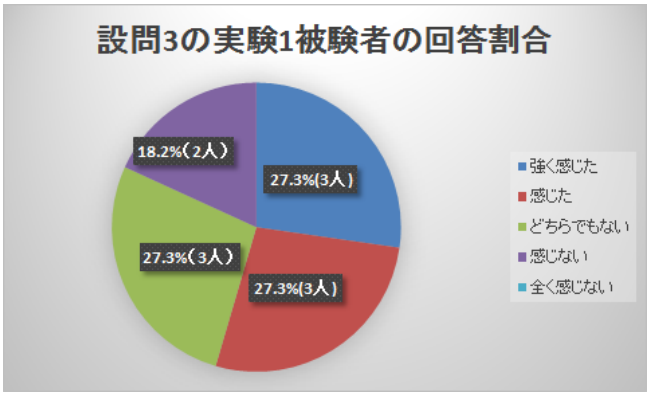

図 7 設問 3 の実験 1 被験者の回答割合

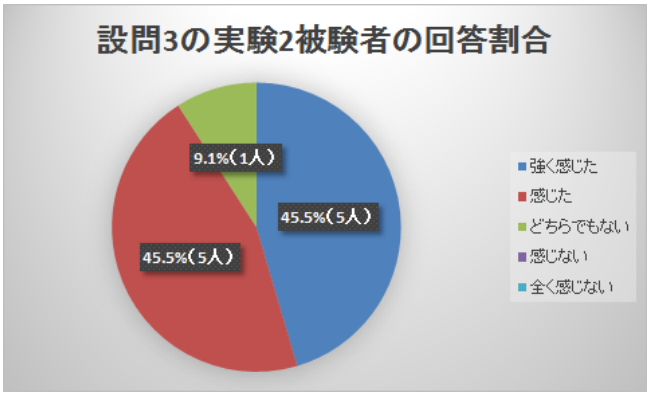

図 8 設問 3 の実験 2 被験者の回答割合

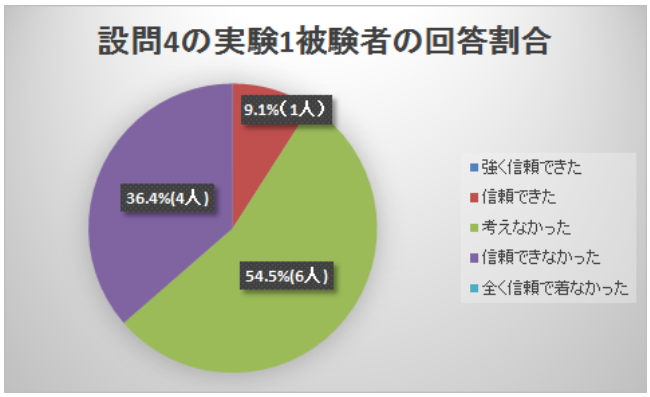

図 9 設問 4 の実験 1 被験者の回答割合

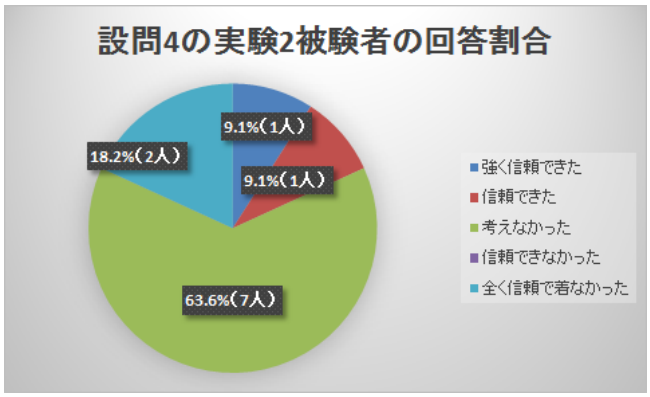

図 10 設問 4 の実験 2 被験者の回答割合

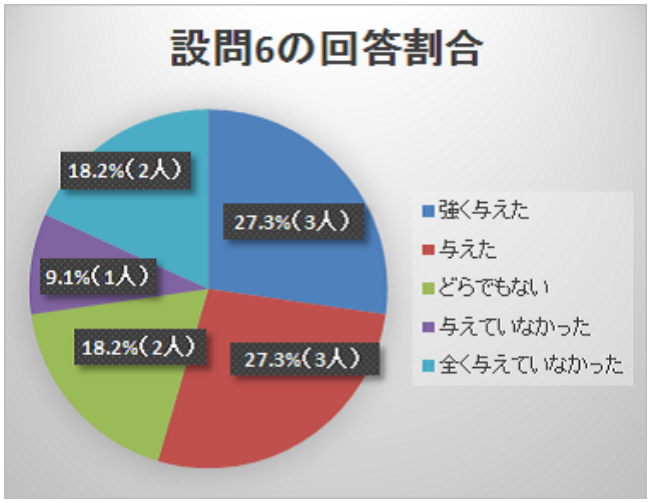

図 11 設問 6 の回答割合

5. さいごに

本論文では、利用されている暗号技術が信頼のできるものであるかどうかをユーザに示すことを目的として、ユーザ側で信頼情報を提示するためのモジュールとそのユーザインタフェースを提案し、暗号技術として検索可能暗号を対象に定め、信頼情報を提示するためのインタフェースとしてウェブブラウザによる表示画面の実装を行った。ユーザに実装した表示画面による信頼情報の提供を行うユーザ

実験を行うことで提案手法を評価した。その結果表示画面を見た 6 名（54.6% ） の被験者より提案手法により信頼が与えられたと回答があり、第 3 者に対する信頼を与えることに寄与していることがわかった。そして実装した表示画面を見た人と見ていない人では実験結果に興味深い差異が生じた。設問 2 において日記の内容を「非常に機微である」または「機微である」と感じた割合は見ていない人が 6 名（54.6% ）なのに対し、見ている人では 9 名（81.9% ）と多い割合を示し、設問 3 における回答においても「強く感じた」あるいは「感じた」と見ていない人では 6 名（54.6% ）なのに対し、見ている人では 10 名（91% ）と多い割合を示したのである。この結果から提案手法は「第 3 者への信頼情報をユーザに提供するという」本来の目的だけではなく「預けている情報のプライバシへの注目」と「第 3 者へデータを預けることへの不安を明らかにする」という 3 点の影響を与える可能性が示された。しかし表示画面が良い評価を得たのは 6 名（54.6% ）であり「第 3 者への信頼情報をユーザに提供するという」本可能性を否定するものではなかったものの高い数字ではなかった。今後は同様の実験をより多くの被験者に対して行うことで、本論文で得られた知見の検証を行い、提案手法が信頼醸成につながるものであるかをより深く調査する。また、今回の提案手法ではウェブブラウザ中のダイアログという手段で提示を行ったが、他の手段での提示による信頼醸成の差を測るなどのアプローチも検討したい。

参考文献

[1] M. Dijk, A. Juels, A. Oprea, R. L. Rivest, E. Stefanov, N. Triandopoulos: *Hourglass Schemes:How to Prove that Cloud Files Are Encrypted*,ACM CCS, 2012.

[2] F. Grobert, C. Willems, T. Holz: *Automated Identifcation of Cryptographic Primitives in Binary Programs*, Lecture Notes in Computer Science Volume 6961, pp 41-60, 2011.

[3] J. Sunshine, S. Egelman, H. Almuhimedi, N. Atri, L. F. Cranor: *Crying Wolf: An Empirical Study of SSL Warning Effectiveness*, Usenix Security Symposium, Montreal, 2009.

[4] D. Akhawe, A. P. Felt: *Alice in Warningland A Large-Scale Field Study of Browser Security Warning Effectiveness*, Usenix Security Symposium, Washington DC, 2013.

[5] R. Curtmola, J. Garay, S. Kamara, R. Ostrovsky: Searchable Symmetric Encryption: *Improved Definitions and Efficient constructions*, Proc. of CCS’ 06, pp.79-88, 2006.