Private Cloud Storage: クライアント側暗号化と安全かつユーザブルなユーティリティ機能の両立

立川 彰宏[1] 金岡 晃[1]

概要：クラウド環境とスマートフォンの発展、セキュリティ・プライバシーの意識の高まりに伴い、エンド間暗号化（End-to-End Encryption、E2E 暗号化）を代表としたクライアント側暗号化がこの 10 年で急速に進んできた。クライアント側暗号化を採用した場合、クラウド側が提供する検索やソーティング、複数端末での利用、他のユーザとのデータ共有といった多岐にわたるユーティリティ機能の活用が制約を受ける。その解決のための手法として、検索可能暗号（Searchable Encryption）や順序付き暗号（Order Preserving Encryption）など、データを暗号化したまま処理が可能な技術に注目が集まっている。しかしこれらを実際にアプリケーションに適用してその有効性を議論した例は少ない。特にユーザビリティの視点でこういった技術が議論されることはほとんどなかった。そこで我々はクラウドストレージに焦点を当て、クライアント側で複数の暗号技術を組み合わせ既存のクラウドストレージサービスと緊密に連携可能な安全かつユーザブルなクラウドストレージを実現するアプリケーションを提案する。そして提案アプリケーションの試作と評価を行い、有用性を示す。我々が提案したアプリケーションは、クライアント側でのファイル暗号化と安全な検索・ソート・他のユーザとのフォルダ共有を実現した。そしてユーザ実験の結果、試作アプリケーションは比較のために開発した非暗号化アプリケーションとのユーザビリティに差が見られないことを示し、提案アプリケーションのユーザビリティが高いものであることを示した。また、実装とユーザ実験を行うことで、コンテンツにクライアント側暗号化を施しつつユーティリティ機能を安全に実現する際の複数の課題が新たに明らかになり、この分野の応用研究に必要性を新たに示すことができた。

Private Cloud Storage: Client-side Encryption and Usable Secure Utility Functions

Akihiro Tachikawa[1] Akira Kanaoka[1]

1. はじめに

利用者はデータをクラウドに持ち、手元のスマホなどの端末でデータの閲覧や処理を行うことが一般的になってきた。LINE や Facebook メッセンジャーなどのメッセージングや Dropbox や Google ドライブなどのクラウドストレージなどが代表的な例である。

クラウドサービスが便利になるとともに、そのセキュリティの重要性が焦点となってきた。その結果多くのサービスでは通信路は TLS/SSL で保護され、エンドユーザとサービス提供者側以外の第 3 者はデータが閲覧できないようになった。さらにプライバシーの重要性も高まることで、サービス提供者に対しても情報が保護されるべきという視点からエンド間暗号化（End-to-End Encryption、E2E 暗号化）を代表としたクライアント側暗号化がこの 10 年で急速に進んできた。いまや主要なメッセージングツールは E2E 暗号化を初期設定時から利用するようになった。

暗号技術は基礎理論から応用研究そして実装を経て社会に大きく展開されるようになった。その結果、E2E 暗号化のように誰しもが利用可能になった暗号技術に対して、あらたな視点での議論が始まるようになった。それがユーザビリティである。暗号技術により利用者データの安全が果たされるが、そのユーザビリティが低いままでは利用者は暗号の利用継続を放棄し、安全な状態は保たれなくなってしまう。暗号化とユーザビリティに関する研究は Whitten と Tygar らによって拓かれ [1]、ユーザブルセキュリティ・プライバシーとして新たな研究分野となった。

クラウドサービスにおける基本機能はデータの送受信であり、そのセキュリティは送受信されるデータの保護が主眼となる。そして暗号化はそのデータ保護を行う基本的な技術となる。一方で、クラウドサービスでは扱うデータを柔軟に処理するために送受信以外の様々なユーティリティ機能を提供している。代表的な機能は検索であろう。また、複数データのソートや複数ユーザ間のデータ共有も主

[1] 東邦大学
Toho University

要なユーティリティ機能といえる。さらにセキュリティの機能としてマルウェアやフィッシングサイトの検出や不適切広告の判定を行うものもある。

クライアント側暗号化を採用した場合、クラウド側が提供する多岐にわたるユーティリティ機能を活用することが難しくなる [2]。その解決のための手法として検索可能暗号（Searchable Encryption）や順序付き暗号（Order Preserving Encryption）など、データを暗号化したまま処理が可能な技術に注目が集まっている。しかしこれらを実際にアプリケーションに適用して有効性を議論した例は少ない。特にユーザビリティの視点でこういった技術が議論されることはほとんどなかった [3]。

そこで我々はそういったユーティリティ機能を豊富に持つクラウドストレージに焦点を当て、Research Question を以下と置いた。

RQ1: クラウドストレージサービスにおけるクライアント側暗号化と安全なユーティリティ機能の同時実現は技術的に可能か？

RQ2: 実現が可能であったとして、そのアプリケーションやサービスはユーザブルであるか？

それらの Research Question に答えるため、本論文ではクライアント側で複数の暗号技術を組み合わせ既存のクラウドストレージサービスと緊密に連携可能な安全かつユーザブルなクラウドストレージを実現するアプリケーションを提案する。そして提案アプリケーションの試作と評価を行い、有用性を示す。

我々が提案したアプリケーションは、クライアント側でのファイル暗号化と安全な検索・ソート・他のユーザとのフォルダ共有を実現した（図 1）。そしてユーザ実験の結果、試作アプリケーションは比較のために開発した非暗号化アプリケーションとのユーザビリティに差が見られないことが示され、提案アプリケーションのユーザビリティが高いものであることを示した。上記により Research Question に答えたことに加え、実装とユーザ実験を行うことでコンテンツにクライアント側暗号化を施しつつユーティリティ機能を安全に実現する際の複数の課題が新たに明らかになり、この分野の応用研究に必要性を新たに示すことができた。

本論文の貢献は以下である。

(1) 既存のクラウドストレージサービスに適用可能な、安全かつユーザブルなクラウドストレージアプリケーション機構の提案とアプリケーションの実現
(2) ユーザ実験による提案アプリケーションはユーザビリティの点で高いことの提示
(3) 安全なユーティリティ機能のより高いレベルでのユーザビリティ実現に向けた複数の課題の発見

本論文の構成は以下の通りである。第 2 章で関連する研究について説明し、第 3 章において提案するクライアント側暗号化と安全なユーティリティ機能の実現の概要を説明する。第 4 章では予備のユーザ実験について述べ、それらの結果を踏まえて行ったユーザ実験の本実験を第 5 章で述べる。第 6 章では得られた結果をもとにした分析結果と考察を示す。第 7 章では本研究や本実験での制限や今後の課題を整理する。最後に第 8 章でまとめる。

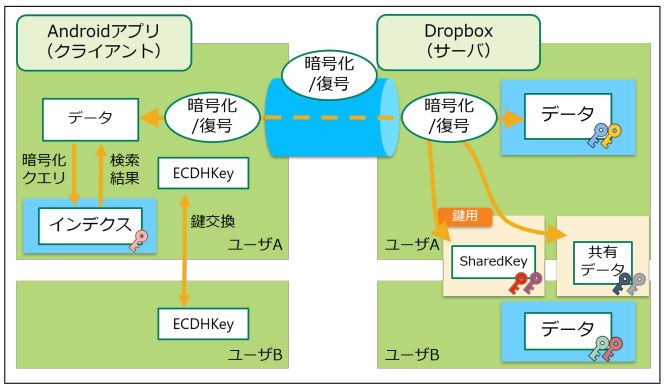

図 1 提案アプリケーションの構成：クラウドストレージサービスのサーバ側機能を変更することなく、クライアント側でコンテンツ暗号化とユーティリティ機能の安全な実現を達成する

2. 関連研究

Whitten と Tygar は初めてセキュリティとユーザビリティについて充実した議論を行った [1]。彼らの研究は電子メールに適用される暗号技術 PGP（Pretty Good Privacy）に対して、安全かつユーザブルに技術が利用可能であるかを調査した。その内容は電子メールに対する暗号技術適用の議論にとどまらず、セキュリティに求められるユーザビリティの要件などを含む広くセキュリティ・プライバシーとユーザビリティを両立するための研究分野を拓くものであった。

暗号化とユーザビリティについての研究はその後 Gerfinkel らによりさらに深く研究がされていき [4], [5], [6], [7] 、Fahl らの Facebook メッセンジャーへの適用 [8] や Sheng らによる電子メールのセキュリティとユーザビリティの両立 [9] など多くの研究が行われてきた。

2013 年に Ruoti らが電子メール暗号化をクライアント側が自動的に行うことで利用者の負荷を軽くする手法を提案するなど、電子メールにおける暗号化とユーザビリティはさらなる高度化を果たした [10], [11], [12]。彼らが提案したシステム Pwm は暗号化処理を完全に自動化をするために「透過的（Transparently）」と表現がされていた。暗号化に関するユーザビリティは、暗号化と復号の作業と、暗号鍵の管理の 2 点に集約されると言ってよい。Pwm はこの両者を自動的にシステム側が行うものであった。その自動化については高いユーザビリティを実現することが期待され実際にその高さが示されたものの、興味深い別の事実も観測された。一部の実験参加者は、暗号化されたときに

はランダム化されたメッセージが暗号文として画面に表示される、という理解をし、それらの表示がなく透過的に暗号化がされた Pwm では混乱を招き逆に暗号化を解除してしまうなどの行動が観測された。ランダムな文字列が表示されることがユーザビリティ確保の一面を担うことは Fahl らの研究でも観測がされていた [8]。

ユーザビリティについての研究はユーザ実験を行うことでその性質を測ることが一般的であるが、背景にある技術や文化の変遷により従来の結果とは異なる動向が実験により得られることもある。Bai らは 2016 年に暗号鍵管理のモデルについて、利用者の認識等のヒアリングを行った [13]。対象となったモデルは利用者自身が鍵を管理する「Key Exchange モデル」と、鍵の管理はサービス提供者側に委託する「Key Registration モデル」の 2 つであった。後者は利用者の負荷は低いがセキュリティ面でリスクが残るモデルであり、利用者はその点の理解に対する調査が焦点であったが、分析の結果は多くの利用者が Key Registration モデルをリスクを認識しつつも受け入れることを示していた。2016 年のこの結果は 2013 年の Ruoti らの結果とは異なるものであり、3 年の間に大きく利用者の認識が変わった可能性を示すものとなった。実際に Ruoti らの研究の実施から Bai らの研究の実施の間に、Facebook Messenger や Whatsapp、LINE など世界的に広く利用されているメッセージングツールで Key Registration モデルを採用した透過的な E2E 暗号化が実施されるようになり [14], [15], [16]、こういった背景も影響したことが考えられる。

電子メールやメッセージングを中心に暗号化とユーザビリティの研究が多く行われているが、クラウドを利用したサービスの中でもう 1 つ重要な位置を占めるクラウドストレージに着目をしてみる。クラウドストレージサービスに対して暗号技術を適用して安全を図るアプローチは、学術的には Kamara らによって開かれた [3]。その後、さまざまな論文が学術的なアプローチで安全なクラウドストレージの実現に向けて発表がされていき、技術の高まりを迎えた [17], [18], [19]。一方で通信路の保護やクラウド側のデータ保護における暗号の利用など、セキュリティやプライバシーを考慮した商用サービスが一般にも広まってきた。TLS/SSL による通信の保護は代表的なサービスである Dropbox、Google Drive、Microsoft Onedrive などで採用がされている。また、サーバ側で保管されている情報を守るためにサーバ側で保管データを暗号化しているケースがある。Dropbox は 128 ビット以上の鍵を用いた AES により暗号化し [20]、Google も保管データを暗号化している [21] としている。

クラウドストレージに関する暗号化は通信路の保護とサーバ側のデータ保護に利用されていて、利用者のデータをサーバ側に秘匿しているわけではない。サーバ側は利用者の情報を見ることが可能である。2016 年、Evernote がプライバシーポリシーの変更を発表した際にこの点が大きな議論となった。Evernote が発表した新しいポリシーでは、新機能開発のために必要に応じて Evernote 側がユーザの情報を確認することが明記されていた。多くの批判を浴び発表の翌日には見直しがされたものの、クラウド側にデータを預ける場合のプライバシーに注目がされた重要なきっかけとなった [22]。

電子メールやメッセージングと同様にクラウドストレージにおいても利用者データのプライバシーを守るためにはクライアント側での暗号化が基本戦略となる。すでに複数の商用サービスでクライアント側の暗号化が行われている [23], [24], [25], [26]。このようにクライアント側で暗号化を行うこと自体は技術的に難しいことではない。問題となるのはその鍵管理とユーザビリティである。学術的なアプローチでこの点を議論した研究はほとんどなく、著者らの知る限り唯一 Fahl らがユーザビリティに焦点を当てたフレームワークとして Confidentiality as a Service（CaaS）を提案していた [27]。CaaS はデータ保護を担う第 3 者サービスであり、クライアント側とクラウドストレージ事業者側の双方の負担を減らすモデルとなっている。プロトタイプとして Dropbox、Thunderbird、Facebook メッセンジャーに適用するアプリケーションやアドオンを試作していた。

一方で、クライアント側で暗号化が行われた場合の制約について議論をしている研究は多くない。緑川らは Pwm のような Web サービスにおいてクライアント側での電子メールの暗号化を行う場合、元の Web サービスの機能が制限される点を指摘した [28]。Pwm の場合、Gmail が提供するメールの検索機能やソート機能が制限される。言及はないものの CaaS についても同様のことが起こりうる。クライアント暗号化においてはそういったサーバ側が提供する検索やソートといった周辺機能（ここではユーティリティ機能と呼ぶ）が制限される可能性が高い。そこで緑川らは Web メールサービス上のユーティリティ機能群の中で検索に焦点を当て、検索可能暗号技術を採用することで安全かつユーザブルに検索が実現できることを示した [28]。

暗号化されたデータに対してユーティリティ機能を提供可能な技術は多く研究が行われている。先ほど言及した検索可能暗号では、共通鍵暗号を利用した対称型検索可能暗号（Searchable Symmetric Encryption）や、公開鍵暗号を利用した検索可能暗号（Publickey Encryption with Keyword Search）が提案されている [29], [30]。SSE は Curtmola らの手法 [31] を中心にさまざまな応用が研究されそのパフォーマンスも実用的なものになってきている [32], [33]。PEKS は Boneh らの手法 [34], [35] から発展しさまざまな機能を持つ手法が多数提案されている [17], [19]。検索以外にも、順序保持型暗号（Order Preserving Encryption）[36], [37] やデータ保持の有無を確認可能な手法 [18]、さらに一般的

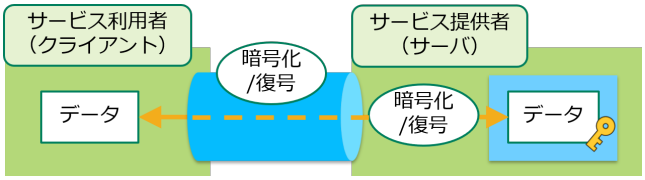

図 2 オンラインストレージサービスのデータ暗号化状況（Dropbox の場合）

な演算をデータを秘匿したまま行うことが可能な完全準同型暗号（Fully-homomorphic Encryption）が提案されている [38], [39]。データベースに特化しさまざまな機能を暗号化されたまま実施可能な CryptDB も提案されている [40]。ユーティリティ機能を安全に実施する手法の研究も進み、いよいよ次の段階としてユーザビリティを併せて検討することが考えられる。

3. クライアント側暗号化と安全なユーティリティ機能の両立

3.1 クラウドストレージサービスにおける脅威のモデル

本研究では、利用者が意図しない情報漏えいに焦点を当ててクラウドストレージサービスの脅威を検討する。

クラウドストレージサービスでは、サービス提供者と利用者、そしてサービスに関係のない第 3 者の 3 つのステークホルダーが想定できる。そして、利用者が意図しない情報漏えいの発生として、以下の 2 つの脅威が挙げられる。

- 第 3 者への情報の漏えい（第 3 者による意図的な情報取得を含む）
- サービス提供者への情報の漏えい（サービス提供者による意図的な情報取得を含む）

第 3 者への情報漏えいとしては、サービス提供者と利用者間の通信路における情報の保護と、サービス提供者側によるデータ保護の 2 種類で対策が可能である。

サービス提供者への情報の漏えいについては、サービス提供者側によるデータの保護だけでは対応できず、利用者側で保護機能の利用をしなければならない。

クラウドストレージサービスを安全に利用するには上述した 2 つの脅威への対応が必要になる。

3.2 システムに求められるユーティリティ機能とその実現方法

サービス利用者の意図しない情報漏えいの 1 つである、サービス提供者への情報漏えいを防ぐためにはクライアント側での情報保護が必要となる。

サービス提供者側にデータをアップロードする際には拡張子を含むファイル名とファイルデータを暗号化して送信し、ダウンロードする際には暗号化されたファイル名とファイルデータをダウンロードしてクライアント側アプリで復号して表示を行うことで、サービス提供者への情報漏えいからデータを守ることができる。ここではファイル名とファイル内容の暗号化を 1 つにまとめて「コンテンツ暗号化」と呼ぶとする。

単純にコンテンツ暗号化を行った場合、サービス提供者側からファイル・フォルダ名やファイル内容などのコンテンツ内容が閲覧できなくなるため、ストレージサービスとして提供されている検索や共有などの機能が正常に機能しなくなる恐れがある。そのため、利用者側でコンテンツ暗号化を行う場合、提供されているいくつかの機能を、暗号化データを考慮した機能へ改良し、利用者側で実装しなければならない。

クラウドストレージサービスで提供される基本的な機能としては、検索、並び替え、データ共有の機能が挙げられる。そこでコンテンツ暗号化に対応させた各機能の対応方法を検する。共有機能に関しては、ファイル共有やフォルダ共有、リンク共有などが存在するが、この研究ではフォルダ共有に着目した。

3.2.1 検索機能

検索の手法は、利用者が検索を行うたびに全文を文字列検索する逐次検索と、あらかじめコンテンツが持っている情報を検索用に準備するインデクス型検索に大別される。逐次検索は事前準備が不要であることから設置が容易である一方で検索の速度は高くない。インデクス型検索はあらかじめ検索用語と結果を保持しているため効率が良い一方で、保護される対象であるコンテンツの情報がインデクス内に解析可能な形で保存される可能性がある。オープンソースの検索エンジンとして代表的な Apache Lucene では、インデクスのデータフォーマットの仕様が公開されており [41]、キーワードやその検索結果が平文のまま保管されているため、インデクス情報からコンテンツの内容が類推できる仕様となっている。本研究では、インデクス保護を実現可能かつ検索効率のよい対称型検索可能暗号（SSE）を採用した。SSE では暗号化インデクスをあらかじめ作成し、検索時には暗号化クエリ（トラップドア）を生成し、暗号化クエリを利用して暗号化インデクスより検索結果を得る。暗号化インデクスはサーバ側とクライアント側のどちらに置いてもよいが、今回は既存のクラウドストレージサービスと緊密に連携可能とするためにクライアント側に置くこととした。

3.2.2 ソート機能

ソート手法は、サーバ側でソートを実施して結果をクライアントに返す方法と、サーバ側にあるデータをダウンロード後に利用者側でソートする方法がある。前者はサーバ側でソートするために、クライアント側暗号化を行うと並び替えが行えない。そこで暗号化したまま並び替えを行うことが出来る手法として、Order Preserving Encryption（OPE）を応用することが可能である。OPE は暗号文同士の順序が保持される暗号であるため、ファイル名や日

時などのソート対象を OPE で暗号化することでサーバ側でのソートは可能となる。ソートの処理性能は OPE のパフォーマンスに依存することとなる。後者はサーバ側にある並び替えに必要なデータをダウンロードし、利用者側で復号を行ってから並び替えを行う必要があるためにデータの量に依存したパフォーマンスとなる。

ユーザが 1 つのフォルダ内に収めるファイル数は利用時の効率を考えると数千や数万になることは考えにくいため、アプリに組み込んだ際のパフォーマンスを考慮し本研究においては後者の方法を採用する。

3.2.3 フォルダ共有機能

クライアント側暗号化を行ったフォルダ共有を実現するためには、暗号化鍵の漏えいなどのセキュリティ面の問題を考慮し、自分のみが閲覧できるファイルに利用している暗号化鍵とは別の共有フォルダ用の暗号化鍵を用意して、共有するフォルダ内のファイルを共有用暗号化鍵によって再暗号化する必要がある。暗号化鍵の付け替え方法は、サーバ側で復号・再暗号化を行う方法と、一度クライアント側に共有予定フォルダ内の暗号化ファイルをダウンロードし、復号・再暗号化を行ってからアップロードし直す方法の 2 種類が存在する。前者はサーバ側で復号することや既存サービスを機能変更が必要となるため、本研究では後者を採用した。

共有フォルダ用の暗号化鍵（SharedKey）は、公開鍵を利用した鍵交換アルゴリズムを利用して共有を行う。暗号化された SharedKey の共有は、共有相手との 1 対 1 の共有をする鍵交換用フォルダをクラウドストレージ上に作成し、暗号化した SharedKey をアップロードし共有することで対応する。被フォルダ共有者側は、鍵交換アルゴリズムを利用して暗号化された SharedKey を復号することで共有フォルダを復号・表示することが出来るようになる。

3.3 試作アプリケーションと利用サービス

3.3.1 アプリケーションの概要

3.2 で提案した機能を実現するクライアント側の Android アプリケーションを試作した。クラウドストレージサービスは Dropbox を選択し、アプリケーションと Dropbox 間の通信などは Dropbox API を介して行う。試作アプリケーションはクライアント側でのコンテンツ暗号化と復号、Dropbox とのデータ通信、ファイル名と更新日時によるソート、SSE を用いた検索、SharedKey の生成と共有の機能を持つ。

ファイルの暗号化には AES を、SSE には尾形らが提案した SimpleSSE[32] を、フォルダ共有を行う際の共有フォルダ用の鍵交換には楕円曲線 Diffie-Hellman 鍵交換（ECDH）を利用した。ECDH の公開鍵のリポジトリは Dropbox とは独立した専用のサーバを用意し、試作アプリケーションに対応している Dropbox アカウントのリストとともに公開鍵を保管する。

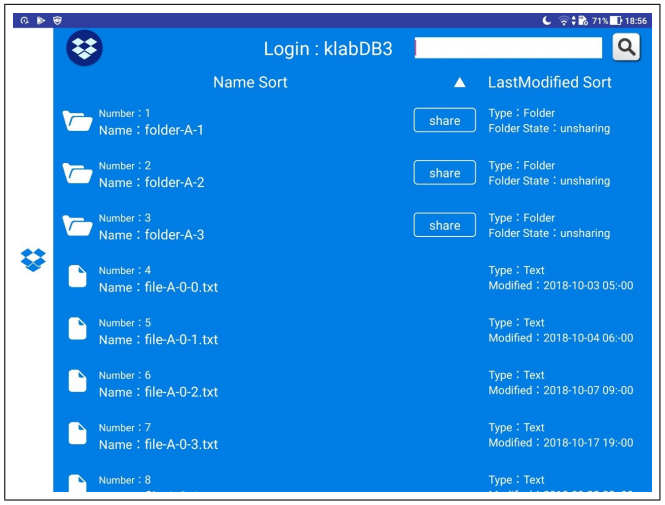

図 3 試作 Android アプリケーションのユーザインタフェース

3.3.2 試作アプリケーションの動作

試作アプリケーションの基本的な動作を解説する。図 3 のように、試作アプリケーションを起動するとファイル名・フォルダ名の一覧と検索窓、ソートボタン、ホームボタンが表示される。フォルダの表示部分にはファイルとは異なり、フォルダ共有ボタンが表示される。

Dropbox に保存されている各ファイルは、ファイルコンテンツ・ファイル名・最終更新日時が暗号化されて Dropbox 上に保存されているため、アプリ側でファイル名を表示する際にファイル名のみ事前にダウンロードし復号して表示している。ファイルはアプリケーションが暗号化を行い Dropbox にアップロードし、新規フォルダは端末側でフォルダ名の暗号化を行い Dropbox 上に作成する。アプリケーション利用者がファイル・フォルダの一覧からファイルのアイコンをタップすると、Dropbox より対象となっている暗号化ファイルをダウンロードし、端末側で復号処理を行った後にファイルに応じたアクションを実行する。フォルダのアイコンをタップすると、Dropbox の対象フォルダの内部にあるファイル・フォルダの一覧の表示を行う。

検索は検索窓にキーワードを入力し検索ボタンを押すことで実行される。検索キーワードを入力し検索ボタンが押されると SSE の暗号化クエリを作成し、アプリケーション内に保管されている暗号化インデクスを使用して検索を行う。そして検索結果から得た暗号化ファイル名を復号し検索結果を表示する。

ソートは、ファイル名と最終更新日時の 2 種類で実施可能とし、それぞれ昇順・降順に並び替える機能を持つ。ソート時は暗号化されたファイル名とフォルダ名を Dropbox からダウンロードし、アプリケーション側で復号した後にソートを行うことで画面に表示する。

共有はフォルダに対して行うことを可能としている。各フォルダの表示位置に共有ボタンを用意し、共有ボタンを

押すことで共有設定が開始する。まずダイアログが開かれ、共有相手の Dropbox アカウントが入力される。入力された Dropbox アカウントが本試作アプリケーションに対応した利用者であるかを確かめるために ECDH 公開鍵リポジトリにアクセスしアカウントを確認する。利用者である場合、その利用者の公開鍵をダウンロードする。次にアプリケーションは共有対象のフォルダのデータを Dropbox からダウンロードし復号化をする。そして SharedKey を生成し、フォルダのデータを SharedKey ですべて暗号化しアップロードする。共有する利用者間で鍵交換用のフォルダがすでに生成されている場合は SharedKey を相手の ECDH 公開鍵で暗号化しアップロードし、生成されていない場合は新たに作成し DropboxAPI を用いて共有設定をしたのちに ECDH により暗号化された SharedKey をアップロードする。そして SharedKey で暗号化されたフォルダに対しても DropboxAPI を用いて共有設定をする。なお、鍵交換用の共有フォルダはアプリケーション画面には表示されないようにしている。

4. 予備実験

4.1 予備実験の実験目的

暗号技術をユーティリティ機能に適用したシステムに対してユーザ実験を行う場合、比較となるシステムの対象は複数考えることができる。たとえば検索機能に絞った場合、以下のシステムが比較対象として考えうる。

(1) クライアント側暗号化なし＋検索機能あり（平文インデクス利用）
(2) クライアント側暗号化あり＋検索機能なし
(3) クライアント側暗号化あり＋検索機能あり（逐次検索）
(4) クライアント側暗号化あり＋検索機能あり（平文インデクス利用）
(5) クライアント側暗号化あり＋検索機能あり（暗号化インデクス利用）

この中で Dropbox 等の既存クラウドストレージは (1) にあたり、本研究で提案したアプリケーションは (5) にあたる。これら 5 つはそれぞれセキュリティとユーザビリティのレベルに違いがあると考えられる。このうち (1)-(4) については差異が考察しやすいが（図 4）、上記 (5) にあたるユーザビリティのレベルが不明であり本研究の目的の 1 つはそこを明らかにすることにある。機能面では (1) と (5) は大きく異なるためにユーザビリティの面で小さくない差が表れて直接的な比較や議論が難しいことも考えられる。そのため十分に評価するためには他のシステムとそれぞれ比較することが必要となる。

また提案アプリケーションが提供するユーティリティ機能は検索だけでなくソート、フォルダ共有と複数にわたる。これらのユーザビリティを統合的かつ十分に評価するためにはより多くの対象システムが必要となる。しかしこれらを網羅的に用意しユーザ実験をすることは容易ではない。

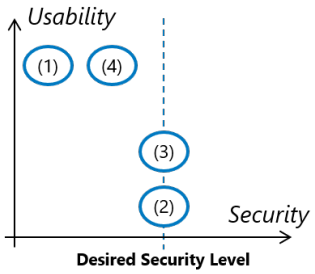

図 4 コンテンツ暗号化と検索機能におけるセキュリティとユーザビリティレベル：4.1 における (1)-(4) の位置づけを経験的な知見をもとにマッピング

そこで本研究では、検索に機能を絞り、提案アプリケーションである (5) がどこに位置するかを明らかにし、(1) と比較評価することが妥当であるかを確かめるための予備実験を行う。

4.2 予備実験の実験概要

この予備実験では、SSE を採用したクライアント側暗号化クラウドストレージアプリケーションのユーザビリティ評価のために、比較対象として機能の異なる 3 種類の Android アプリを作成し計 4 種類の Android アプリを利用してユーザ実験を行った。ユーザ実験は System Usability Scale（SUS）で用いられる質問子そのアンケート回答に対しての半構造化インタビューを採用した。

実験中の行動は動画撮影し、インタビュー内容と動画内容から Grounded Theory Approach（GTA）を行うことで質的な分析を行った。SUS の質問子には、SUS の原文の 10 項目を和訳したものを使用した。

4.3 予備実験の実験手順

今回の予備実験では、本来の研究目的を伝えることで偏ったユーザ行動が観察されてしまうことを避けるために、本来の研究目的とは異なる仮の実験目的を立てた。予備実験は、最初に仮の実験目的とタスクの説明をした後に実際に Android アプリケーションを用いて実験参加者にタスクを実施してもらう。タスク終了後に本当の実験目的を伝えたのちに実験協力の同意を得て、SUS アンケートと関連質問によるインタビューを行った。

タスクは「利用者の方がパスワードによってロックをかけたフォルダが存在するが、利用者の方がそのパスワードを忘れてしまったので、本人確認を行うために 4 桁の数値を入力しなければならない」という状況のもとで、その 4 桁の数字がある 4 つのキーワードによるアプリケーション内での検索結果数を並べたものだと教え、実際に 4 桁の数字の取得をしてもらうこととした。

インタビューでは、SUS 項目の各回答の理由を問い、またアプリ利用におけるストレスの実感と要因、検索の効率、

OS	Android 6.0
対応 CPU	Qualcomm Snapdragon 650（8 コア）
対応メモリ	4GB
ストレージ機能	eMMC：32GB

表 1 実験で使用した Android 端末の基本性能

AppID	タイプ	検索機能	コンテンツ暗号化	インデクス暗号化
App1	(1)	○	×	×
App2	(2)	×	○	−
App3	(3)	○	○	−
App4	(5)	○	○	○

表 2 予備実験で使用したアプリについて

オンラインサービス利用におけるセキュリティの意識をヒアリングした。

実験で利用する検索対象のファイルには、実験参加者がファイルの中身を見た際に目視で単語を見つけ出すことが難しくなるような文章にするために、10 種類の英文の童話をテキストファイルで用意した。

また、本人確認を行う際の与えられた 4 つの単語については、一般的な用語であり利用した童話にも少なからず出現する可能性を考慮して、色の名称を 4 種類を指定した。

4.4 予備実験で使用した端末とアプリケーション

実験で使用する Android 端末の基本性能を表 1 に示す。

予備実験では機能の異なる 4 つの Android アプリを用意し、それぞれのユーザビリティを測定、機能の違いによりユーザビリティに変化が生じるかを観測する。予備実験に使用する 4 種類の Android アプリについての情報を表 2 にまとめた。そこにはタイプとして 4.1 節で示した分類も記載した。4 つのアプリケーションのうちどのアプリケーションを使うかはランダムに決め、実験参加者には 4 種のアプリケーションがあることとどのアプリケーションを使っているかの双方とも伝えずに実施した。4 つの Android アプリの外観については、外観によるユーザビリティの差が生じないように同一にした。

App1 は、既存の Dropbox のアプリとほぼ同等の機能を持つアプリとなっている。App2 は、コンテンツ暗号化・復号のみを行い検索機能がないアプリとなっている。App3 では検索機能が与えられているがインデクスを使用しておらず、Dropbox 内の暗号化ファイルを全てダウンロードし、全ファイルに対して復号の処理を行った後に、全文の文字列一致により検索を行うものとなっている。App4 はコンテンツ暗号化に加え検索機能を SSE で提供するアプリとなっている。

4.5 実験参加者の募集方法と実験参加者情報

実験参加者の募集は、2017 年 6 月 5 日〜2017 年 8 月 25 日に著者らが所属する大学のシステムと学内掲示板にて行った。実験参加者は計 11 人集まり、そのうち 5 名が男性で 6 名が女性であった。いずれも理学部に所属する学生であり、9 人が情報科学科、2 人が化学科所属であった。

報酬は複数の視点から検討の上で 500 円の図書カードとした。いくつかのユーザ実験を行っている論文を見る限り、アメリカ合衆国の論文では報酬が 1 時間当たり 10 ドル程度であったが [10]、アメリカ合衆国での最低賃金が 15 ドルへ引き上げられた後から 15〜20 ドルへと変化しており [11], [12], [13]、その国の 1 時間当たりの最低賃金を基準とした報酬額であることが見て取れた。そのため、千葉県の最低賃金が実験当時の時間給が 842 円であり [42]、実験予定時間が 30 分であったため、最低賃金の時間給の半分を超える 500 円が妥当であると考えた。

矛盾	使いやすい・便利
ストレス	効率がよい
面倒ではない	面倒・大変・手間・複雑
難しい・分からない	理解・自信・簡単
既存サービスとの比較	分からない（回答放棄）
暗号	安全・セキュリティの意識
統合性	説明書を読む
説明書を読まない	（分類不可能）

表 3 予備実験に対する GTA の結果得られた 16 の概念

4.6 生命倫理審査委員会の承認

実験を行うにあたり学内の生命倫理審査委員会に承認を得て実験を行った。

4.7 予備実験の実験結果と考察

4.7.1 実験結果概要

予備実験を行った結果、クライアント側暗号化と SSE 利用について他のアプリとの間に特筆すべき差異はなく、少なくとも SSE の適用についてはユーザビリティの問題点は確認されなかったと言える結果が得られた。

分析結果等の実験結果の詳細情報についてを以下に示す。

4.7.2 GTA を行ったうえで分類された概念の考察

実験で得られたアンケート回答・実験行動（総数：188（重複含む））を基に GTA を行い、計 16 の概念に分類を行った。

得られた概念の一覧を表 3 に示す。

それぞれの概念において特徴的なものを分析した結果を示す。

いずれの概念について抽出した実験参加者のコメントも、アプリに依存した差異が認められるものが少なく、提案アプリケーション利用のユーザビリティは決して低いものではないことがわかった。たとえば、「使いやすい・便利」に分類されたコメントでは、検索機能を持つアプリに対する肯定的意見が並び、その方法やパフォーマンスについての言及はされていなかった。「効率がよい」に分類さ

れたコメントでは App2 以外の実験参加者によるコメントは分類されず、検索機能の欠如が関係していることがこちらも見て取れた。

もし提案したアプリケーション機構に近い App4 のユーザビリティが低い場合には、「面倒・大変・手間・複雑」への分離や「ストレス」に分類にネガティブな内容が含まれることが予想されたが、コメントの内容と利用アプリケーションを見ると、そのストレス性や面倒さは検索機能がない App2 に対するものが多く、App4 に限定してコメントが多かったようなものはなかった。

4.7.3 主実験で比較する対象の考察

これらの結果から 4.1 における (5) に該当する App4 のユーザビリティは決して低いものではないことが考えられる。そのため主実験では本来の目的である (1) と (5) の比較に焦点を当てて行うことが妥当と考えた。

5. 主実験

5.1 実験目的

予備実験で得た結果をもとに、検索機能とソート機能、そしてフォルダ共有機能を持った提案アプリケーションの評価を行う。

5.2 実験概要

主実験では予備実験の際の分析方法と同じく、クライアント側暗号化クラウドストレージサービスのユーザビリティ評価のために比較対象として機能の異なる Android アプリケーションを作成し、計 2 種類の Android アプリを利用して、SUS のアンケート回答に対しての半構造化インタビューを軸としたユーザ実験を行う。また、実験所要時間とユーザビリティの関係性を観測するため、実験時に実験参加者自身に実験の各項目の所要時間を計測してもらった。

SUS の回答をスコアリングすることで実験結果を量的に分析し、実験中の行動を動画撮影しインタビュー内容と動画内容から GTA を行うことで質的な分析を行う。SUS の質問項目には予備実験のサイト同じく、SUS の原文の 10 項目を和訳したものを使用した。

予備実験の際に用意した仮の実験目的に関しては、偏ったユーザ行動が観察されることを防止するために用意したが、一部の実験参加者の混乱を招く結果となった。そのため主実験においては、実験参加者の実験の目的を先に伝えることとした。なお実験参加者が利用する比較アプリが 2 種類存在していることとどちらのアプリケーションを利用しているかの通知をしないことは予備実験と同じく伝えない。

5.3 実験手順

主実験は、実験目的とタスク説明をした後にタスクを実行してもらい、その後に SUS アンケートと関連質問によるインタビューを行った。

複数のユーティリティ機能のユーザビリティを分析するために、実験参加者のタスクは複数用意した。またいくつかのタスクは実験担当者による指示のもと作業を実施した。

i) ファイル内容の確認：実験実施者により指示を受けたファイルのオープンと内容確認
ii) フォルダ内ファイルのソート：実験実施者により指示を受けたフォルダの更新日時順のソートと最新のファイル名の回答
iii) キーワードによる検索：実験実施者が指示したキーワードで検索を行い、検索結果件数の回答
iv) フォルダ共有 1：実験実施者が実験参加者とフォルダ共有を実施し、実験参加者は共有されたフォルダ内容を確認
v) フォルダ共有 2：実験実施者が指示したフォルダに実験参加者がフォルダ共有を実施し、実験実施者は共有されたフォルダ内容を確認

なおそれぞれのタスクに要した時間を実験参加者自身が別の Android 端末を用いて計測を行った。

インタビューでは、SUS 項目の各回答の理由を問い、またアプリ利用におけるストレスの実感と要因、オンラインストレージのアプリとしての効率、オンラインサービス利用におけるセキュリティの意識、実験用アプリを利用した際の実行時間について感じたことをヒアリングした。

実験で利用するファイルは予備実験時と同様に、著作権フリーの英語テキストファイルを用意し、実験参加者用・実験実施者用のファイルだと一見して理解できるようにして利用した。

検索キーワードに関しては予備実験の際の選出方法と同様に、一般的な用語であり、選出したファイルにも少なからず出現する可能性を考慮して色の名称を選択し、その中から 3 種類を選出した。

5.4 主実験で使用するアプリケーション

主実験では予備実験の結果を踏まえ、コンテンツのクライアント側暗号化に加えて検索とソート、フォルダ共有の 3 つのユーティリティ機能にそれぞれ暗号技術を適用した Android アプリケーション CryptApp と、同様のユーティリティ機能を暗号技術を適用せずに利用するアプリケーション PlainApp の 2 種類を用意した。2 つのアプリケーションの外観については、外観によるユーザビリティの差が生じないように同一にしてある。

CryptApp は、クライアント側でのコンテンツ暗号化に加え、検索・ソート・フォルダ共有の機能をクライアント側暗号化に対応させたアプリとなっている。検索機能は SimpleSSE を利用して実装しており、検索インデクスは端末のローカルストレージに保管している。ソート機能は、ファイル名一覧を Dropbox からダウンロードした後に復

実験参加者 ID	学科	性別	利用アプリ	SUS スコア
P1	情報科学科	女性	CryptApp	92.5
P2	情報科学科	男性	CryptApp	75.0
P3	情報科学科	男性	CryptApp	87.5
P4	情報科学科	男性	CryptApp	87.5
P5	情報科学科	女性	CryptApp	72.5
P6	情報科学科	男性	CryptApp	90.0
P7	情報科学科	男性	PlainApp	87.5
P8	情報科学科	女性	PlainApp	57.5
P9	情報科学科	男性	PlainApp	70.0
P10	情報科学科	女性	PlainApp	95.0
P11	情報科学科	男性	PlainApp	82.5

表 4 主実験の実験参加者情報

実験 No	実験項目	PlainApp 平均（s）	CryptApp 平均（s）
i	ファイル内容確認	10.338	10.178
ii	ソート	18.687	21.708
iii	キーワード検索	11.826	12.837
iv	フォルダ共有 1	33.027	53.813
v	フォルダ共有 2	24.599	49.474

表 5 各実験項目ごとの実験所要時間の平均と比較

号し、端末側でソートを行い表示する。フォルダ共有の機能は、共有フォルダを SharedKey で暗号化し、ECDH を利用し SharedKey を交換し共有フォルダ用の暗号化鍵を受け渡すことで実現した。

5.5 実験参加者について

実験参加者の募集は、2018 年 12 月 17 日〜2019 年 1 月 25 日に大学の履修システムと学内掲示板により行った。

実験参加者の情報を表 4 に示す。ここには後述する SUS のスコアも付記した。主実験の実験報酬は予備実験で設定した報酬と同じく 500 円分の図書カードを報酬とした。

5.6 生命倫理審査委員会の承認

主実験を行うにあたり学内の生命倫理審査委員会に承認を得て実験を行った。

6. 実験結果と考察

6.1 実験所要時間の集計結果

主実験で得られた各実験項目ごとに実験参加者全体の実験所要時間の平均時間を計算し、時間比較を行ったものを表 5 にまとめた。

各実験項目ごとの平均と比較の結果を見ると、暗号化アプリと非暗号化アプリで最も大きく差が開いた項目は実験 4 と 5 のフォルダ共有の実験であった。表を見ると暗号化アプリのフォルダ共有にかかる時間が非暗号化アプリに比べて、20 秒以上差が出ていることが確認できる。2 種類のアプリ間で 20 秒以上差が開いたのは、暗号化処理の部分ではなく暗号化アプリの共有処理中に行われるダウンロードとアップロードの処理が主な原因であると考える。暗号化アプリでフォルダ共有を行う際、共有フォルダの暗号化鍵交換のために一度 Dropbox から共有ファイルをダウンロード・復号し、共有フォルダ用の暗号化鍵で暗号化してからアップロードという処理を行う。このダウンロード・アップロードの処理以外は非暗号化アプリと同様に DropboxAPI を利用した処理を行っているため大きな差は生まれにくいと考えられる。

6.2 SUS スコアの集計結果

各実験参加者から得られた SUS スコアは表 4 に示されている。主実験で得られた SUS のスコアのアプリケーションごとの平均を取った結果、PlainApp は 78.5、CryptApp は 84.17 となった。

6.3 GTA により分類された概念の考察

主実験で得られたアンケート回答・実験についての質問（総数 : 154）を基に GTA を行い、計 9 の概念に分類を行った。それぞれの概念において特徴的なものを分析した結果を示す。

6.3.1 機能理解

この概念に含まれる回答は、大きく分けて「アプリケーションの操作方法の理解」と「アプリケーションを操作する前の事前知識の理解」の 2 種類に分類される。操作方法の理解に関しての回答は、さらに 2 種類の意見に分かれており、

> 「シンプルなのでパッと見で分かる。」（*P4*、*CryptApp*）
> 「分かりやすいと思ったのでどんな人でも利用できると思う。」（*P6*、*CryptApp*）
> 「他のアプリと似た操作方法なのですんなり使い方が入ってくる。」（*P10*、*PlainApp*）

という様に操作方法がシンプルでどんな人でもすぐに利用できるという意見と、

> 「この手のアプリを利用したことがあればサポートは要らないと思う。」（*P6*、*CryptApp*）
> 「タップした先の動作やログインなどの機能については高齢者の方には説明が必要だと思う。」（*P7*、*PlainApp*）
> 「パソコンやスマホを利用している人だったら使えると思う。」（*P10*、*PlainApp*）

という様に PC やスマートフォンの利用経験によってはサポートが必要となるのではないかという意見が存在した。

アプリケーションを操作する前の事前知識の理解に関しての回答は、特に事前知識は必要ないと回答する実験参加者が過半数を占めていたが、

> 「英語以外は問題ないと思った。」（*P1*、*CryptApp*）

「そこそこ英語が分かれば分かると思う。」（*P8*、*PlainApp*）
「クラウドに関する知識やフォルダ共有に関して学ぶ必要があると思う。」（*P9*、*PlainApp*）
「多くの事は学ぶ必要はないが、フォルダなどの概念については学ぶ必要があると思う。」（*P7*、*PlainApp*）

とフォルダやクラウドの機能について学ぶ必要があると考える実験参加者も存在した。暗号化に関する知識が必要だと回答した実験参加者は 1 名のみであり、

「暗号化、シェア周りの知識は若干必要かも。シェア機能で暗号化を無為にしてしまうなどの知識は身に付けた方がいいかもしれない。」（*P4*、*CryptApp*）

とフォルダ共有を行うことで暗号化機能が利用できなくなることを考慮していた。

6.3.2 効率

この概念には、クラウドストレージの効率についての回答が分類された。この概念に含まれる回答の多くは前向きなものだった。

「シェアとかがやりやすかったので効率は良いと思う」（*P6*、*CryptApp*）
「効率は良かったと思う。」（*P11*、*PlainApp*）

効率が良くなかった・悪かったという悪印象な意見は、どちらのアプリケーション利用者からも確認されなかったものの、以下のように比較材料が不十分なために比較ができないと回答した実験参加者もいた。

「普通が分からないので効率がいいか分からない。」（*P5*、*CryptApp*）
「私自身がこのファイルを入れたわけではないので今回の実験では分からなかった。」（*P10*、*PlainApp*）

6.3.3 処理時間

この概念には、実験で使用したアプリケーションの実行時間についての回答が分類されている。CryptApp を利用した実験参加者 6 名の回答としては、

「若干フォルダ共有まわりが遅かったと感じた。それ以外の部分は特に感じなかった。」（*P3*、*CryptApp*）
「普通のアプリと比べると処理が長い。具体的にはシェア機能が若干遅かった。それ以外の部分は特に待ち時間が長いとは感じなかった。」（*P4*、*CryptApp*）

という様にフォルダ共有の処理時間が遅かったと感じた実験参加者と、

「早すぎず、遅すぎずだったと思う。特に何も感じなかった。」（*P1*、*CryptApp*）
「特に何も感じなかった。」（*P2*、*CryptApp*）
「実行時間については特に何も感じなかった。」（*P5*、*CryptApp*）
「比較的普通。あまり遅くもなく普通に動いてるんじゃないかなと思った。」（*P6*、*CryptApp*）

という様に特に処理時間について何も感じなかったと回答している実験参加者の双方が存在した。CryptApp 利用者すべてに共通する点として、フォルダ共有部分の処理以外は特に何も感じていないという事が挙げられる。

PlainApp を利用した実験参加者 5 名の回答としては、

「フォルダ共有の部分がそこそこ時間がかかってる気がした。気になった部分はフォルダ共有のみ。」（*P8*、*PlainApp*）
「少しだけファイルを開く際のスピードが遅いかなと思った。普段のアプリだとこのアプリの *2/3* くらいのスピードで開ける気がする。通信速度の問題かもしれないけど。」（*P9*、*PlainApp*）
「シェアの時だけ少し待つのかなって気がしたけど、こんなものなのかなという感じだった。」（*P10*、*PlainApp*）

と CryptApp と同じくフォルダ共有の処理が遅かったと感じた実験参加者が複数いた。また CryptApp を利用した実験参加者からは出なかった「ファイルを開く速度」について言及する実験参加者も存在した。

6.3.4 ストレス

この概念には、アプリケーションを利用した際に感じたストレスに関しての回答が分類されている。分類された回答のほとんどは「ストレスは感じなかった」という回答であったが、

「全部英語で出てきちゃうのでパッと見どこを押せばいいのか分からない。英語の部分が使いづらかった。」（*P8*、*PlainApp*）
「英語だったのでちょっと面倒に感じた。」（*P8*、*PlainApp*）
「英語で表記されている部分にストレスを感じた。それ以外は特に感じなかった。」（*P8*、*PlainApp*）

とアプリケーション UI やファイル名・ファイル内容が英語表記されてたことで大きくストレスを感じたと回答する実験参加者が存在した。英語表記されていること以外でストレスを感じたと回答している実験参加者は確認されなかったため、アプリケーションの機能に関してのストレスを感じた実験参加者は存在しなかったことが伺える。

6.3.5 セキュリティ・暗号の意識

この概念には、セキュリティの意識と暗号化についての回答が分類されている。セキュリティの意識については、実験を行う前まではクラウドストレージのセキュリティを意識したことが無かった実験参加者がほとんどであった。セキュリティを意識したことがある実験参加者は、

「怪しいファイルをダウンロードしたらどうしようとか。」（*P1*、*CryptApp*）

「手元にデータがないのでデータが消えたり、中身をいじくられたりという可能性は感じていた。」（*P4*、*CryptApp*）

「ダウンロードする際に安全なのかなと思ったことはある。利用者側としてこちらの情報が漏れてしまうかなと考えたことはない。」（*P11*、*PlainApp*）

とサービス提供者側に閲覧される脅威に関して意識したことがある実験参加者はほとんどおらず、唯一サービス提供者側に閲覧される脅威に関して意識したことがあると回答した実験参加者の回答は、

「少し意識したことがある。インターネットに挙げられているものが本当に完全に他の人に見られていないのかなとか。提供している側に見られているのは多少あるのかなと思っていた。危機は感じていない。」（*P2*、*CryptApp*）

と可能性は考えていたものの脅威としては認識していなかったという回答であった。

暗号化については、

「簡単なセキュリティだと中身が見られてしまうから暗号化は必要だと思った。」（P2、CryptApp）

「Dropbox を使ったことがあるが、特に変わりがないので安全性が高いなら使いたい。」（P3、CryptApp）

「運営の方に閲覧されてしまうという説明を聞いて、今回利用した暗号化されてるアプリを利用させてもらって不自由がなかったので、見られるよりは見られないやつをという事で使いたいと思う。」（P6、CryptApp）

と暗号化されているのであれば利用したいといった意見がほとんどであり、機能面が既存のサービスと変わりないのであれば暗号化を施したアプリケーションを利用したいと考える実験参加者が多くいることが確認された。

6.3.6 使い勝手（アプリケーション全般）

この概念には、アプリケーションの使い勝手に関する回答が分類されている。この概念に分類された回答はほとんどがアプリケーションに対して肯定的な意見であり、「使いやすかった」「シンプルだった」「分かりやすかった」と両方のアプリケーション利用者から意見が出ていることから全体を通して使いやすいアプリケーションになっていることが伺える。

「シンプルで検索もわかりやすくてすごく使いやすかった。」（*P7*、*PlainApp*）

「情報を共有するときとか、普段は使用しないが使うならこのアプリで十分使えると思った。」（*P11*、*PlainApp*）

「機能の使い方が分かりやすかったので使いやすいと思う。」（*P6*、*CryptApp*）

「普段と同じように利用できるので統合されているのかなと思った。」（*P9*、*PlainApp*）

6.3.7 使い勝手（機能）

この概念には、機能面を重視したアプリケーションの使い勝手に関する回答が分類されている。実験参加者の機能に対する回答は賛否の双方があったがであった。機能に対する否定的な意見としては、

「一見して更新ボタンが分かりづらかった。」（P11、PlainApp）

「背景が全部同じ色なので、ソートボタンがチュートリアルがないと分かりづらい。」（P10、PlainApp）

「ファイル名の表記がどこからか判断するのが少し大変だった。」（P7、PlainApp）

とアプリ UI に関する意見がほとんどであり、クラウドストレージの機能に関しては特に言及がなかった。対して肯定的な意見としては、

「フォルダを共有できるのはいいな。」（P1、CryptApp）

「ソート、フォルダ移動とか復号とかがうまく統合されていると思った。」（P4、CryptApp）

「ソートや検索など必要な機能が全部ある。」（P8、PlainApp）

「普段利用している Dropbox と同じ感覚で利用できたので使いやすいと思った。」（P9、PlainApp）

と既存の Dropbox のアプリケーションと同じように利用できるという意見が存在し、クラウドストレージ機能に関しては使い勝手が悪いと回答している実験参加者は確認されなかった。

6.3.8 矛盾

どちらのアプリケーション利用者からも「特に矛盾は感じなかった」「矛盾だと思うものは無かった」という意見を出していることから、アプリケーションの機能や挙動に対して矛盾を感じていないことが伺えた。

6.4 主実験結果に対する考察

SUS スコアによる差に特筆すべきところはなく、GTA の分析結果からは暗号技術を適用したユーティリティ機能やクライアント暗号化についてユーザビリティ視点での大きな否定はなかった。否定的な部分があった面は暗号化を利用していない PlainApp についても言及されていることから、それらは提案アプリケーションの機能によるものではなく、クラウドストレージやそのユーティリティ機能が本来持つ面に依拠していると考えられる。

7. 獲得した知見と今後の課題

提案アプリケーションの試作時や実験時、GTA 分析時において、研究目的とは異なる視点ではあるが今後の研究において解決すべきと考えられる課題がいくつか明確になった。本章ではそれらを整理する。

7.1 ユーティリティ機能の暗号技術適用にかかわる課題

ユーティリティ機能に暗号技術を実装する際に解決すべき課題をまとめた。今後、この分野が進むにつれて解決が欠かせない課題であると考えられる。

7.1.1 ファイル移動時のインデクス変更への対応

検索インデクスを用いて検索を行う場合、検索対象のファイルを移動するとインデクスを参照した際の検索結果のファイルパス情報を変更する必要がある。クラウドストレージのようにデータが手元に存在しない場合にクライアント側でインデクスを利用した検索を行う場合に、オンラインストレージ上のデータの移動を常に監視しリアルタイムでインデクスに反映させる必要が出てくるため、クライアント側に負担になる可能性がある。

7.1.2 共有時・共有解除時のインデクス編集

クラウドストレージでファイルやフォルダの共有を行うと、共有を受けた利用者は他のユーザのデータ閲覧が可能になる。そのため、自身でファイルやフォルダを追加した際と同じく共有されたファイルやフォルダも検索インデクスに追加する必要が出てくる。インデクスへの追加は、自身がデータを追加した時と同様にインデクス追加することが出来るが、共有解除時には単純な仕組みでは実現が難しい。共有を設定した相手ユーザから共有を解除された場合、インデクスから該当するファイルやフォルダの情報を削除すべきであるが、インデクス自体からそれを判断し適切にインデクスを削除することは難しい。たとえばリアルタイムでオンラインストレージ上のデータを監視するなどが考えられるが、削除されたファイルやフォルダにかかわる情報だけをインデクスデータから適切に削除できるのかという点も課題となる。

7.1.3 共有開始時の承認タイミングでの共有データ情報提示

一般的にクラウドストレージサービスの共有機能は、共有したい相手に共有申請を送り、被共有者が共有相手と共有するデータを確認してから共有承認を行うことで共有が成立する。

本研究のモデルでは、共有開始時に表示されるファイルやフォルダ名は共有承認前であるために SharedKey の共有が完了しておらず、ファイルやフォルダ名を復号することができない。被共有者がデータ名を閲覧できるのは、共有申請を許可し共有データにアクセスが可能な状態になってからであるため、被共有者はデータ名が不明瞭なまま共有申請を許可しなければならない。

解決方法としてデータ名のみを別の仕組みで暗号化と復号を行い共有相手に提示することが挙げられるが、承認タイミングでの共有データ情報の提示は部分的な情報の漏えいとなるため、どの程度の情報漏えいまで許容するかという課題が出てくる。

7.1.4 共有時の鍵管理

本研究で実装したアプリケーションでは 1 フォルダ共有ごとに 1 共有鍵の作成を行うため、利用を進めるうちに共有鍵の数が多くなり、鍵管理の負担が高くなる。共有フォルダと共有鍵の対応表管理が必要となるなど、鍵管理が重要になると考えられる。

7.2 クラウドストレージサービス仕様上の課題

クラウドストレージサービスに特有の課題もいくつか明らかになった。今回の試作アプリケーションで利用した Dropbox を利用する中で明らかになった課題を説明する。

7.2.1 共有解除の仕様

DropboxAPI の仕様では、共有解除を行う際にホスト側（共有設定側）とゲスト側（被共有設定側）で異なった内部処理が行われている。

ホスト側が共有解除を行う場合、指定したフォルダが共有フォルダから通常フォルダへ戻り、フォルダ共有を行っているメンバー全員が共有解除される。一方で、共有を受けたゲスト側が共有解除を行う場合、指定したフォルダから自身が共有解除される（アクセスが不能になる）だけで処理が終わるため共有を行っている残りのメンバーに影響は出ない。ホストとゲストが 1 対 1 で共有を行っていた場合、ゲスト側で共有解除を行うとホストだけが共有メンバーとなっている共有フォルダが出来てしまう。この場合、DropboxAPI は共有解除をこの場合、ホストは共有しているメンバーが他に存在しないがホスト自身の鍵とは異なる鍵で暗号化したフォルダを持つこととなる。鍵管理の複雑さが増加すると考えられる。そのため、暗号との連携の際にはホスト側・ゲスト側で異なる検討が必要だと考える。

7.2.2 共有フォルダへのファイル追加時の処理

共有フォルダへファイルが追加されたときに Dropbox では明示的にクライアント側にイベント発生が通知されないため独自のイベントハンドラが必要となる。

7.3 サービス化を行う上での課題

この節には実験用の試作アプリケーションでは実装出来ているが、実際にサービスを行う上では改良が必要となると考えられる課題をまとめた。

7.3.1 SharedKey 共有用フォルダ

試作アプリケーションではフォルダ共有を行う際に共有フォルダ用の鍵を共有するために共有相手と 1 対 1 で共有する SharedKey 共有用フォルダを別途用意している。SharedKey 共有用フォルダは 1 対 1 で共有を行うため、ホストは共有者の数だけの SharedKey 共有用フォルダを持つことになる。その際のフォルダの名付け方や容量の問題が課題となると考えられる。

7.3.2 パス名全体のインデクス化

検索結果のファイルへアクセスするためには、Dropbox

上のパスを指定する必要がある。試作アプリケーションでは Dropbox 上のパス名まで含めたファイル名を検索結果としてインデクスに保存している。このように検索結果にファイル名だけではなくファイルの位置情報であるパス名までを含める場合、検索結果の表示や検索結果のファイルへアクセスする際の暗号化対応処理の複雑さが増加する。Dropbox 上のファイルを操作した際のインデクス処理の複雑さが課題となる。

7.4 制約

本研究で提案したアプリケーションや実施した実験にはいくつかの制約がある。ここではそれらを整理することで、本研究で明らかにしたことをより明確にすることを狙う。

7.4.1 実験参加者数

主実験の実験参加者は 11 人であり、そのうち PlainApp 利用が 5 人、CryptApp 利用が 6 人と、多い実験参加者数とは言えない実験であった。Nielsen らによると、5 人でユーザ実験を行うことでユーザビリティ上の問題の 85% が発見できるという分析がされている [43]。本研究で行った実験はセキュリティに関するものであり、またプラットフォームとして Android を利用した実験であるため、必ずしも Nielsen らの実証結果と一致するとは限らず、より多い実験参加者数が望ましいものの実験の実験参加者数が足りないということはないと考える。

7.4.2 複数環境での鍵共有

クラウドストレージの大きな利点の 1 つは、1 人の利用者が複数の環境で同じデータを閲覧・編集可能であることがある。クライアント側暗号化を施したクラウドストレージに対しても同様の利点が期待されるところであるが、その場合には利用者の鍵を複数環境に安全に配付する仕組みが必要になる。本研究で提案したアプリケーションのモデルでは複数環境への安全な鍵配付についての考慮は検討外としている。

対策方法としては、Bai らが調査した研究 [13] で調査対象となった Key Registration モデルの採用が考えられる。たとえばクラウドストレージに利用者鍵を収めるフォルダを用意し、各環境からアクセスする場合にはまずそのフォルダにアクセスして鍵を取得する方法である。この場合、単に鍵をクラウドストレージにアップロードするだけでは鍵の安全性が保たれないために保護する必要がある。その保護機構として、パスワードにより暗号鍵を生成するパスワード導出関数の利用や、FIDO 規格にあるような生体認証やハードウェアトークンによる認証の利用が考えられる。

8. まとめ

本研究では、クラウドストレージサービスにおけるクライアント側のデータ保護に焦点を当て、クライアント側の暗号化と安全なユーティリティ機能実現のためのアプリケーションの提案を行った。提案アプリケーションは既存のクラウドストレージサービスに直接適用可能でありながらコンテンツ暗号化だけでなく検索・ソート・共有を安全に実現可能とした。提案アプリケーションの試作を用いたユーザ実験では、暗号技術を適用することによるパフォーマンス低下によるユーザビリティ影響がみられないことを示し、提案アプリケーションが安全かつユーザブルなクラウドストレージ環境を実現可能であることを示した。

参考文献

[1] Alma Whitten and J. D. Tygar. Why johnny can't encrypt: A usability evaluation of pgp 5.0. In *Proceedings of the 8th Conference on USENIX Security Symposium - Volume 8*, SSYM'99, pp. 14–14, Berkeley, CA, USA, 1999. USENIX Association.

[2] 緑川達也, 金岡晃. ジョニーはまだ暗号化できない？：暗号化とユーザビリティに関する研究の調査. 情報処理学会論文誌, Vol. 59, No. 12, pp. 2120–2131, dec 2018.

[3] Seny Kamara and Kristin Lauter. Cryptographic cloud storage. In Radu Sion, Reza Curtmola, Sven Dietrich, Aggelos Kiayias, Josep M. Miret, Kazue Sako, and Francesc Sebé, editors, *Financial Cryptography and Data Security*, pp. 136–149, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[4] Simson L. Garfinkel and Robert C. Miller. Johnny 2: A user test of key continuity management with s/mime and outlook express. In *Proceedings of the 2005 Symposium on Usable Privacy and Security*, SOUPS '05, pp. 13–24, New York, NY, USA, 2005. ACM.

[5] Simson L. Garfinkel, Jeffrey I. Schiller, Erik Nordlander, David Margrave, and Robert C. Miller. Views, reactions and impact of digitally-signed mail in e-commerce. In *Proceedings of the 9th International Conference on Financial Cryptography and Data Security*, FC'05, pp. 188–202, Berlin, Heidelberg, 2005. Springer-Verlag.

[6] Simson L. Garfinkel. Enabling email confidentiality through the use of opportunistic encryption. In *Proceedings of the 2003 Annual National Conference on Digital Government Research*, dg.o '03, pp. 1–4. Digital Government Society of North America, 2003.

[7] Simson L. Garfinkel, David Margrave, Jeffrey I. Schiller, Erik Nordlander, and Robert C. Miller. How to make secure email easier to use. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '05, pp. 701–710, New York, NY, USA, 2005. ACM.

[8] Sascha Fahl, Marian Harbach, Thomas Muders, Matthew Smith, and Uwe Sander. Helping johnny 2.0 to encrypt his facebook conversations. In *Proceedings of the Eighth Symposium on Usable Privacy and Security*, SOUPS '12, pp. 11:1–11:17, New York, NY, USA, 2012. ACM.

[9] Steve Sheng, Levi Broderick, Colleen Alison Koranda, and Jeremy J Hyland. Why johnny still can't encrypt: evaluating the usability of email encryption software. In *Symposium On Usable Privacy and Security*, pp. 3–4, 2006.

[10] Scott Ruoti, Nathan Kim, Ben Burgon, Timothy van der Horst, and Kent Seamons. Confused johnny: When automatic encryption leads to confusion and mistakes. In *Proceedings of the Ninth Symposium on Usable Privacy and Security*, SOUPS '13, pp. 5:1–5:12, New York, NY, USA, 2013. ACM.

[11] Scott Ruoti, Jeff Andersen, Scott Heidbrink, Mark O'Neill, Elham Vaziripour, Justin Wu, Daniel Zappala, and Kent Seamons. "we're on the same page": A usability study of secure email using pairs of novice users. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pp. 4298–4308, New York, NY, USA, 2016. ACM.

[12] Scott Ruoti, Jeff Andersen, Travis Hendershot, Daniel Zappala, and Kent Seamons. Private webmail 2.0: Simple and easy-to-use secure email. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, pp. 461–472, New York, NY, USA, 2016. ACM.

[13] Wei Bai, Moses Namara, Yichen Qian, Patrick Gage Kelley, Michelle L. Mazurek, and Doowon Kim. An inconvenient trust: User attitudes toward security and usability tradeoffs for key-directory encryption systems. In *Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)*, pp. 113–130, Denver, CO, 2016. USENIX Association.

[14] N. Unger, S. Dechand, J. Bonneau, S. Fahl, H. Perl, I. Goldberg, and M. Smith. Sok: Secure messaging. In *2015 IEEE Symposium on Security and Privacy*, pp. 232–249, May 2015.

[15] P. Rösler, C. Mainka, and J. Schwenk. More is less: On the end-to-end security of group chats in signal, whatsapp, and threema. In *2018 IEEE European Symposium on Security and Privacy (EuroS P)*, pp. 415–429, April 2018.

[16] Takanori Isobe and Kazuhiko Minematsu. Breaking message integrity of an end-to-end encryption scheme of line. In Javier Lopez, Jianying Zhou, and Miguel Soriano, editors, *Computer Security*, pp. 249–268, Cham, 2018. Springer International Publishing.

[17] Z. Xia, X. Wang, X. Sun, and Q. Wang. A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data. *IEEE Transactions on Parallel and Distributed Systems*, Vol. 27, No. 2, pp. 340–352, Feb 2016.

[18] C. Chris Erway, Alptekin Küpçü, Charalampos Papamanthou, and Roberto Tamassia. Dynamic provable data possession. *ACM Trans. Inf. Syst. Secur.*, Vol. 17, No. 4, pp. 15:1–15:29, April 2015.

[19] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou. Privacy-preserving multi-keyword ranked search over encrypted cloud data. *IEEE Transactions on Parallel and Distributed Systems*, Vol. 25, No. 1, pp. 222–233, Jan 2014.

[20] Architecture - dropbox business.

[21] Security - google cloud help.

[22] Evernote revisits privacy policy change — evernote — evernote blog.

[23] Spideroak secure software — spideroak.

[24] Encryption software to secure cloud files — boxcryptor.

[25] Cryptomator: Free cloud encryption for dropbox & others.

[26] End-to-end encrypted cloud storage for businesses — tresorit.

[27] S. Fahl, M. Harbach, T. Muders, and M. Smith. Confidentiality as a service – usable security for the cloud. In *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 153–162, June 2012.

[28] T. Midorikawa, A. Tachikawa, and A. Kanaoka. Helping johnny to search: Encrypted search on webmail system. In *2018 13th Asia Joint Conference on Information Security (AsiaJCIS)*, Vol. 00, pp. 47–53, Aug 2018.

[29] Christoph Bösch, Pieter Hartel, Willem Jonker, and Andreas Peter. A survey of provably secure searchable encryption. *ACM Comput. Surv.*, Vol. 47, No. 2, pp. 18:1–18:51, August 2014.

[30] Geong Sen Poh, Ji-Jian Chin, Wei-Chuen Yau, Kim-Kwang Raymond Choo, and Moesfa Soeheila Mohamad. Searchable symmetric encryption: Designs and challenges. *ACM Comput. Surv.*, Vol. 50, No. 3, pp. 40:1–40:37, May 2017.

[31] Reza Curtmola, Juan Garay, Seny Kamara, and Rafail Ostrovsky. Searchable symmetric encryption: Improved definitions and efficient constructions. In *Proceedings of the 13th ACM Conference on Computer and Communications Security*, CCS '06, pp. 79–88, New York, NY, USA, 2006. ACM.

[32] Wakaha Ogata, Keita Koiwa, Akira Kanaoka, and Shin'ichiro Matsuo. Toward practical searchable symmetric encryption. In Kazuo Sakiyama and Masayuki Terada, editors, *Advances in Information and Computer Security*, pp. 151–167, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[33] Fei Han, Jing Qin, and Jiankun Hu. Secure searches in the cloud: A survey. *Future Generation Computer Systems*, Vol. 62, pp. 66 – 75, 2016.

[34] Dan Boneh and Matt Franklin. Identity-based encryption from the weil pairing. In Joe Kilian, editor, *Advances in Cryptology — CRYPTO 2001*, pp. 213–229, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[35] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public key encryption with keyword search. In Christian Cachin and Jan L. Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004*, pp. 506–522, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[36] Rakesh Agrawal, Jerry Kiernan, Ramakrishnan Srikant, and Yirong Xu. Order preserving encryption for numeric data. In *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*, SIGMOD '04, pp. 563–574, New York, NY, USA, 2004. ACM.

[37] Alexandra Boldyreva, Nathan Chenette, Younho Lee, and Adam O'Neill. Order-preserving symmetric encryption. In Antoine Joux, editor, *Advances in Cryptology - EUROCRYPT 2009*, pp. 224–241, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[38] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing*, STOC '09, pp. 169–178, New York, NY, USA, 2009. ACM.

[39] Abbas Acar, Hidayet Aksu, A. Selcuk Uluagac, and Mauro Conti. A survey on homomorphic encryption schemes: Theory and implementation. *ACM Comput. Surv.*, Vol. 51, No. 4, pp. 79:1–79:35, July 2018.

[40] Raluca Ada Popa, Catherine M. S. Redfield, Nickolai Zeldovich, and Hari Balakrishnan. Cryptdb: Protecting confidentiality with encrypted query processing. In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*, SOSP '11, pp. 85–100, New York, NY, USA, 2011. ACM.

[41] Apache lucene - index file formats.

[42] Chiba minimum wages table — togane-shi homepage.

[43] Jakob Nielsen and Thomas K. Landauer. A mathematical model of the finding of usability problems. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, CHI

'93, pp. 206–213, New York, NY, USA, 1993. ACM.