全文検索エンジン Apache Solr における API を用いた平文推測攻撃の評価

藤原 啓成[†1]　鈴木 貴之[†1]　吉野 雅之[†1]

概要：近年，個人情報保護に関する法規制対応向けの製品・サービスに暗号化ストレージが利用されつつある．一方，法規制対応向けに監査責任者が個人情報等の機微情報を含む文書を特定する際に，組織内文書を網羅的に検索可能なよう全文検索インデックスは平文であるため，全文検索サーバの管理者が閲覧可能となる．本稿では，全文検索インデックスが平文の場合，サーバの管理者が暗号化ストレージ内の文書を推測可能であることを明らかにする．一般の管理者が解析可能な Apache Solr を対象に，内部 API を用いて取得可能な全文検索インデックスの情報を再構成し，高確率で個人情報を含む文書を復元可能と考えられる結果を報告する．

キーワード：個人情報保護，全文検索インデックス，内部不正，全文検索エンジン，Apache Solr

Evaluation of plaintext guessing attack by using full-text search engine Apache Solr API

Keisei Fujiwara[†1]　Takayuki Suzuki[†1]　Masayuki Yoshino[†1]

Abstract: In recent years, encrypted storage is being used for the regulatory compliance products and services, relating to protection of sensitive information. On the one hand, responsible persons for the audit of their organization is obliged to comprehensively manage documents including sensitive information such as personal information for compliance with laws and regulations. Therefore, a full-text search service that aggregates in-house documents is being provided. However, since the full-text search index in the service is processed in plaintext, it can be viewed by the administrator of the full-text search server. In this paper, we clarify that the administrator can guess documents in encrypted storage when the full-text search index is plaintext. For Apache Solr that can be analyzed by general administrators, we reconstruct the information of full-text search index that can be acquired using internal API and report the results that are considered to be able to restore documents containing sensitive information with high probability .

Keywords: Personal information protection, Full-text search index, Insider attack, Full-text search engine, Apache Solr

1. はじめに

近年，国内外の個人情報保護に関する法規制対応向けの製品・サービスに暗号化ストレージが利用されつつある．一方，法規制対応向けに監査責任者が個人情報等の機微情報を含む文書を特定する際に，組織内文書の機微情報を網羅的に検索可能なよう全文検索インデックスは平文のままである. そのため, 全文検索インデックスの情報はすべて，全文検索サーバの管理者が閲覧可能となる．本稿では，暗号化ストレージを利用しても全文検索インデックスが平文の場合は，サーバの管理者が暗号化ストレージ内のドキュメントを推測可能であることを明らかにする．本稿では，特別な知識を有さないサーバの管理者でも解析可能であることを示すため, 全文検索エンジン Apache Solr[1]を対象に，全文検索インデックスの管理用の内部 API を用いて，取得可能な全文検索インデックスの情報を再構成し，高確率で機微情報を含む元の文書を復元可能と考えられる結果を報告する．

2. 全文検索インデックス

2.1 全文検索インデックス

全文検索インデックスとは, 全文検索エンジンにおいて，検索対象の文書から抽出した文字列と，抽出元の文書のリンクを対応付けて，検索しやすい形に構造化したデータである．全文検索エンジンは，利用者からの検索クエリの文字列に対応する文書リンクのリストを，全文検索インデックスを検索することで特定し，利用者へ応答する(図 1).

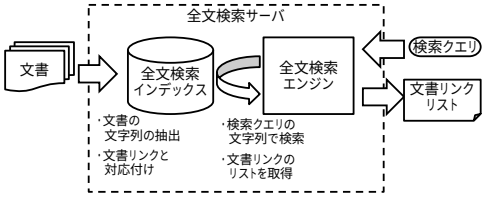

図 1　全文検索インデックスの位置づけ

Figure 1　Position of Full-text search index

†1 株式会社日立製作所研究開発グループ
Hitachi, Ltd. Research & Development Group

文字列	文書リンク(ID)	位置情報(オフセット)
ABC	1	15
		100
	3	40
DE	4	50
・・・	・・・	・・・

図 2　Apache Solr の全文検索インデックス情報
Figure 2　Full-text Search Index of Apache Solr.

Apache Solr の全文検索インデックスは，文字列と文書の対応情報，文書中における各文字列の位置情報（オフセット）を含む(図 2). 全文検索サーバは，文書中のテキスト情報から，形態素解析や N-gram などの手法で文字列を抽出し，全文検索インデックスとして格納する．形態素解析の場合，特殊記号等の検索に適さない文字列を除く文字列が全文検索インデックスに格納される．一方, N-gram の場合，１文字ずつずらして，N 文字ずつ文字列を切り出し，重複を含む文字列が全文検索インデックスに格納される．こうした手法により，文書が含むほぼ全ての文字列が，全文検索インデックスに格納される．

Apache Solr 等の全文検索エンジンは，これらの情報を用いて, 利用者の検索クエリに対応する文書リンクを特定し，そのリストを利用者に応答する．さらに，全文検索エンジンは，全文検索インデックスの情報を利用することで，文書の一部を文書リンクと共に表示するハイライト機能等の拡張情報を利用者へ提供している．

全文検索エンジンは，利用者が外部から参照可能な Web API と，サーバ内部で全文検索インデックスから情報を抽出する内部の API を備える(図 3). 主要な全文検索エンジンの OSS である Apache Solr の場合は，内部に全文検索インデックスを管理する OSS である Apache lucene[2]を備え，全文検索インデックスの操作に Apache lucene の API を用いる．Apache Solr は，Web API で利用者からの検索クエリを受け付け，全文検索サーバの内部で Apache lucene の API を用いて全文検索インデックスから検索クエリに対応する情報を検索し，Web API の出力情報として構成して，利用者へ応答する[1][2][3].

本稿では，Apache lucene の API など，全文検索サーバの内部で利用される全文検索インデックスの情報を処理する API を内部 API と呼ぶ.

なお，Apache Solr と並ぶもう一つの主要な全文検索エンジンの OSS である Elasticsearch は，Apache Solr と同様に，内部に Apache lucene を使用している.

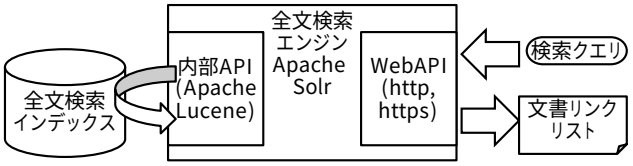

図 3　Apache Solr の Web API と内部 API
Figure 3　Web API and inner API in Apache Solr

2.2 課題

近年広まっているクラウドサービスの利用において，個人情報や営業機密等の機微情報の安全管理措置・機密保持のため，クラウドサービスに文書を格納する場合，アクセス制御や暗号化ストレージが利用されつつある.

一方で，2018 年 5 月に施行される GDPR 対応等において，個人情報を扱う組織が，個人の要求に応じて個人情報を参照・移転・削除可能とするため，組織内文書内のある個人情報を含む文書を特定することが必要となってきている．このため，GDPR ソリューションを提供するクラウドサービサーは，ソリューションメニューに全文検索サービスを組み込んでおり，これを利用することで監査対象となるすべての組織内文書が検索可能となる[4].

しかし，通常はファイル単位ではアクセス制御や暗号化で参照範囲等が限られているデータが，クラウド内の全文検索サーバに集約される．全文検索サーバの管理者は，全文検索エンジンの内部 API を用いて，全文検索インデックスに集約された機微情報を含む組織内文書のデータにアクセス可能となる(図 4).

本稿では，こうしたデータの集約により，クラウド内の全文検索サーバ管理者が攻撃者となった場合に，脅威となりうる内部 API を用いた平文推測攻撃による文書の再構成の可能性を評価する.

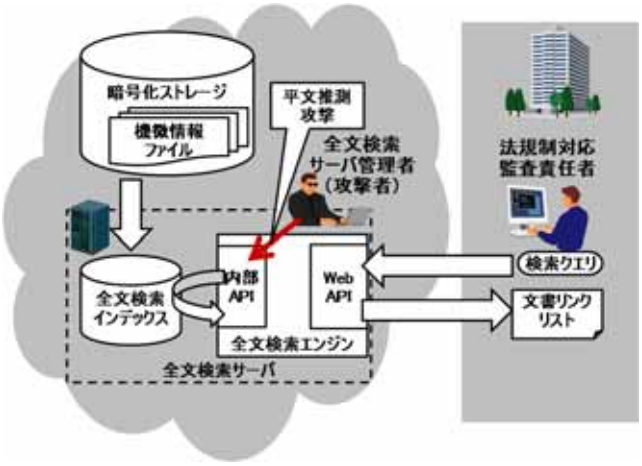

図 4　全文検索インデックスへの平文推測攻撃の脅威
Figure 4　Threat of plaintext guessing attack to full-text search index

以下，想定する脅威についてより詳細に説明する．

(1) **前提**

- クラウドストレージを使って情報保管
- 機微情報ファイル本体は機密保持のため暗号化
- 監査対応に全文検索が必要なため，全文検索インデックスは平文
- 全文検索サーバ管理者は内部 API を使用できる

(2) **攻撃**

- 全文検索サーバ管理者が攻撃者で，機微情報の取得を目的としている
- ユーザ（ファイルの保有者）が暗号化かつアクセス制御されている前提で預けていた機微情報ファイルが，内部 API を用いる平文推測攻撃により，全文検索サーバ管理者が機微情報を復元し取得される．

3. 平文推測攻撃方法

3.1 Apache Solr の全文検索インデックスの構成ファイル

Apache Solr は，特定のディレクトリに一連の全文検索インデックスの構成ファイル(拡張子.doc,.fdt,.fdx,.pos,.tim 等)を保存する．ディレクトリ内には，全文検索インデックスを構成する文字列情報，文書 ID 情報，文字列の位置情報等が複数のファイルに分けて保存されている．各ファイル内の情報は，文字列の情報等を除き，ファイルの状態では圧縮・バイナリ化されている等，可読性が低い．全文検索エンジンが各ファイルをロードし，内部 API からアクセス可能することで情報を容易に取得可能となる．

3.2 内部 API

全文検索サーバ管理者が使用できる主要な Apache Solr の内部 API を示す．

(1) **文書 ID 一覧の取得**

全文検索インデックスが含む文書 ID の一覧を出力する．

(2) **構成情報のタイプ(field)情報の一覧の取得**

全文検索インデックスにおける構成情報のタイプ(field)情報の一覧を取得する．この構成情報のタイプ(field)名は，利用者が定義でき，一例としてはファイル名を格納する“ title ”という field 名や，ファイルの本文を格納する“ text ”などの field 名が定義される．

(3) **文字列と文書 ID の関連付け一覧の取得**

全文検索インデックスにおける構成情報のタイプ(field)名を入力とし，そのタイプの構成情報として全文検索インデックスが含む文字列の一覧と，各文字列を含む文書 ID の一覧を取得する．

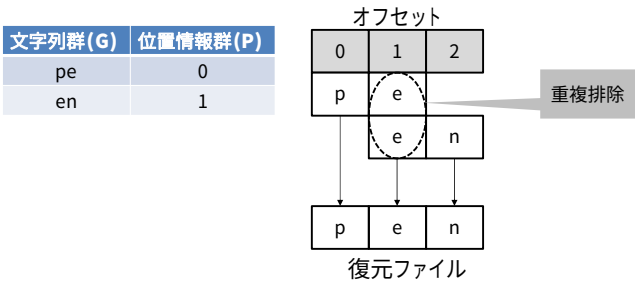

図 5　同一オフセットの複数文字の重複排除処理

Figure 5　Deduplication of same offset characters

(4) **文字列の位置情報（オフセット）の取得**

ある文字列を入力とし，その文字列がどの文書 ID の文書の先頭から何番目（何バイト目）に位置するかの情報を取得する．

3.3 処理手順

全文検索サーバ内で，開発者が利用可能な全文検索エンジンの内部 API を利用した平文推測攻撃の処理手順を示す．ここでは，特定の文書 ID (X)を持つ１つのファイルの，ファイルの本文(field 名:text)を，各内部 API を用いて復元する手順を示す．

(1) **特定の文書 ID (X)が含む文字列群(G)の取得**

「文字列と文書 ID の関連付け一覧の取得」API に,ファイルの本文の構成情報のタイプ（field 名:text）を入力し，文字列と文書 ID の一覧 (Y)を取得する．次に，(Y)から，文書 ID(X)を持つ文字列をすべて抽出したものを，文字列群(G)とする．

(2) **文字列群(G)の文書 ID (X)内の位置情報群(P)を取得**

「文字列の位置情報（オフセット）の取得」API に，(G)の各文字列を入力し，得られる出力から，文書 ID(X)に関する位置情報群(P)を取得する．

(3) **文字列群(G)と位置情報群(P)を用いた本文の再構築**

文字列群(G)の各文字列を，位置情報群(P)のオフセットの小さい順に，復元ファイルの先頭から順に格納する．この際, 同一のオフセットに複数の同じ文字が重なる場合は，重複排除した上で復元ファイルへ出力する(図 5).

3.4 仮定するセキュリティモデル

本稿では，クラウドのデータセンタ内の全文検索サーバ管理者を攻撃者とする．攻撃対象は，クラウドの利用者がクラウド上の暗号化ストレージに保存している組織内文書のうち，GDPR 等の個人情報保護規制対応のために全文検索サービスに登録した組織内文書の全文検索インデックス

である．攻撃者は，全文検索エンジンの内部 API を利用して，全文検索インデックスから情報を抽出し，全文検索サーバ上で処理することができる．全文検索インデックスのデータ構造は Apache Solr を想定し，3.3 節の処理手順により復元ファイルを取得する．攻撃者は，主要な OSS の API を利用する復元プログラムを Java やシェルスクリプト等の主要な開発言語により開発できるスキルを備える．また，攻撃者は平文情報の推測を攻撃手段とし，それ以外の攻撃(破壊，サービス停止等)を考えない．

4. 評価

4.1 評価方法

本稿では，特別な知識を有さないサーバの管理者でも解析可能であるかを評価するため，主要な OSS である全文検索エンジン Apache Solr を対象に，Apache Solr の内部 API を用いた 3.3 節の処理手順による評価プログラムを開発し，平文推測攻撃を実施した．平文推測攻撃による復元データの元データに対する再現率は，元データの文字数に対し，復元データが一致する文字数の割合とした．

評価用文書ファイルとしては，32KB(10,207 文字(空白除く))の NDA 契約に関する機密文書を模したサンプルの文書ファイルを用いた(図 6)．全文検索インデックスとしては，平文状態の全文検索インデックスに加え，全文検索インデックスが暗号化されていた場合の攻撃可能性を評価するため，〝文字列〟カラムを暗号化した全文検索インデックス情報を用いた．

また，現実的な処理時間で攻撃できる否かを評価するため，評価プログラムの実行時間を評価した．評価環境は，PC(CPU:2.4GHz[4core],Memory:8GB,Disk:SSD,OS:Windows 10)上で動作する Virtual Machine(CPU:[2core],Memory:4GB, OS:CentOS)に，Apache Solr 及び評価プログラムを導入した．

図 6　評価用文書ファイル(NDA サンプル)の一部

Figure 6　A part of test file (NDA sample)

表 1　平文推測攻撃の評価結果

Table 1　Evaluation results of of plaintext guessing attack

全文検索 インデックスの種類	再現率(%)	実行時間(秒)
全て平文	96.9	0.38
"文字列"カラムが暗号文	1.3	1.71

4.2 結果

平文推測攻撃の評価結果を表 1 に示す．

平文推測攻撃による再現率は，全文検索インデックスがすべて平文の場合 96.9%であった．再現率の具体的な算出方法は，元データの文字数（空白を除く）を L，元データにはあるが復元データには無い文字数（空白を除く）を M とすると，次の式で算出した．

再現率（%）＝100×(L-M)/L

この場合の，具体的な復元ファイルの一部を図 7 に示す．復元ファイルは，“ (” や”.”等の記号を除く，ほぼ全ての文字列を可読な状態で含んでいる．

一方，全文検索インデックスの “ 文字列 ” カラムが暗号の場合の再現率は 1.3%であった．この場合，平文と同様の算出方法を用いると，暗号文が多様な文字を含むことから，上記 M が極めて少なくなるものの，復元データは可読性がほぼ無いという矛盾した状態となる．そのため，この場合の再現率は，元データが含む 2 文字以上の文字列のうち復元データに含まれる文字列の文字数を N とし，次の式で算出した．

“ 文字列 ” が暗号文の場合の再現率（%）＝100×N/L

この場合の，具体的な復元ファイルの一部を図 8 に示す．復元ファイルは，暗号文の文字列の中に，元ファイルと一致する”is”や”use”など 2 文字あるいは 3 文字の単語を部分的に含むが，可読性は無い．なお,“ .”となっている部分は，暗号文を構成するビット列のうち，対応する文字コードが無いものが “ . ” と表示されている．

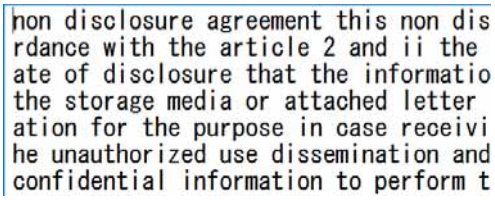

図 7　全て平文の場合の復元ファイルの一部

Figure 7　A part of recovery file from all plaintext index

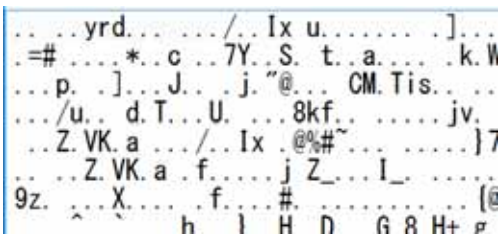

図 8“文字列”カラムが暗号文の場合の復元ファイルの一部

Figure 8　A part of recovery file from partial encrypted index

また，評価プログラムの実行時間は，すべて平文の場合 0.38 秒,“ 文字列 ”カラムが暗号文の場合 1.71 秒であった．実行時間は，いずれも 10 回測定した結果の平均値である．なお，今回の評価では，それぞれの場合で，全文検索インデックスに格納されていたファイル数・サイズが異なる．全て平文の場合は，４ファイル登録・全文検索インデックスのディレクトリサイズは 444KB であるのに対し,“ 文字列 ” カラムが暗号文の場合は，152 ファイル登録・全文検索インデックスのディレクトリサイズは 9.2MB である．このため,“ 文字列 ”カラムが暗号文の場合の実行時間は，暗号化による文字列の肥大化に加え，登録ファイル数・インデックスサイズがより多い状態での測定結果である．

4.3 考察

本稿の平文推測攻撃により，全て平文の全文検索インデックスに対して，96.9%という高い再現率の復元ファイルを生成できたことから，元データの文字列が高確率で再現可能であると考えられる．なお，再現されなかった文字列は，Apache Solr が評価対象文書から全文検索インデックスを生成する際に，検索キーワードとして適さない助詞等の文字列を省くことが主な原因である．

また，平文推測攻撃用のプログラムは，全て平文の場合には 1 秒以内という現実的に実行可能な処理時間で実行できた．このことから，Apache Solr に類似したインデックス構造を備える他の全文検索エンジンでも，現実的な処理時間で平文推測攻撃が実行可能であると考えられる．

さらに，全文検索インデックスの〝文字列〞カラムを暗号化した場合，本稿の平文推測攻撃方法では，ほぼ文書を再現できないことが分かった．検索可能暗号等の技術を用いて，〝文字列〞カラムを暗号化したまま検索処理可能とすることができれば，本稿のセキュリティモデルにおいては平文推測攻撃を防ぐことができると考えられる．ただし，大量の組織内文書が登録される全文検索エンジンでは，大量のマッチング処理が発生すると考えられるため，高速処理が可能な検索可能暗号技術・製品が適すると考える[5][6][7].

5. おわりに

本稿では，全文検索インデックスに対する平文推測攻撃を評価した．全文検索エンジン Apache Solr を対象に，標準的な開発スキルを備えるサーバ管理者が攻撃可能な内部 API を用いて，現実的な処理時間で平文推測攻撃が可能であることを示した．さらに，全文検索インデックス内の文字列カラムが暗号文の場合，本稿の平文推測攻撃では元の文書が再現できないことを示した．文字列カラムを暗号化する対策を実施する場合，大量の組織内文書が登録される全文検索エンジンでは，大量のマッチング処理が発生すると考えられるため，高速処理が可能な検索可能暗号技術・製品が適する[5][6][7].

参考文献

[1] “Solr”. http://lucene.apache.org/solr, (参照 2017-08-25)
[2]McCandless, Hatcher. Lucene in Action 2nd. Manning,2010, 488p
[3]山田浩之，末永匡．検索エンジン自作入門．技術評論社，2014，222p.
[4] “druva ライフサイエンス向けデータ侵害緩和”. http://jp.druva.com/solutions/life-sciences, (参照 2017-08-25).
[5] Steven Zittrower, “Encrypted Phrase Searching in the Cloud”, Global Communication Conference (GLOBECOM), 2012 IEEE, p.764-770.
[6] K.Sakiyama and M. Terada, “Toward Practical Searchable Symmetric Encryption”, IWSEC2013, LNCS8231, pp.151-167, 2013.
[7] “Credeon Secure Full-text Search”. http://www.hitachi-solutions.com/securesearch/, (参照 2017-08-25)