Androidアプリが自動的に収集するデータの暗号化ラベルの実態調査

桜庭 侑星[1,a)]　稲吉 弘樹[1]　齋藤 彰一[2]　門田 暁人[1]

概要：スマートフォンを利用する際，アプリがデータを平文で送信した場合，第三者が通信を傍受し，機密情報や個人情報が漏洩する危険性がある．2022年7月にGoogle Playにデータセーフティセクションが導入され，アプリ開発者に対して，データ送信時の暗号化状況の開示（暗号化ラベル）が求められた．ユーザへ安全なアプリが提供されるためには，まずラベルが正しいこと，そしてラベル上は暗号化しない場合であってもプライバシーデータは暗号化されていることが必要である．しかし，このラベルはアプリ開発者の自己申告制であり，また，これまでの調査も不十分のため，これらの実態は明らかになっていない．本稿は，アプリが自動的に収集するデータにフォーカスして，暗号化ラベルを調査する．国内のGoogle Playからダウンロードしたランキング上位6,180アプリをAndroid端末上で実行し，通信を収集する．アプリと同時にダウンロードしたラベルを，アプリの通信データと比較する．結果，ラベルに反して実際には暗号化しないアプリを11個と，ラベル上は暗号化しないとあり，実際にプライバシーデータを平文で送信するアプリを12個発見した．

キーワード：Android，データセーフティセクション，送信データ暗号化，プライバシー漏洩

Investigation of Data Encryption Labels Corresponding to Data Automatically Collected by Android Apps

Yusei Sakuraba[1,a)]　Hiroki Inayoshi[1]　Shoichi Saito[2]　Akito Monden[1]

Abstract: If data is transmitted without encryption on a smartphone, confidential or personal information can be leaked to a third party. In July 2022, Google Play introduced a data safety section requiring app developers to disclose the encryption status when transmitting data (i.e., encryption labels). The label must be correct, and privacy data must be encrypted even if the label indicates no encryption. However, these labels are self-reported by app developers, and there has been no sufficient investigation to date. This paper investigates encryption labels corresponding to data automatically collected by apps. The top 6,180 apps collected from Google Play in Japan were run on Android devices, and communication data were collected. We compared the labels with the apps' communication data and found 11 apps that, contrary to their labels, do not actually encrypt data, and 12 apps that state they do not encrypt data on their labels but actually transmit privacy data in plain text.

Keywords: Android, Data safety section, Network data encryption, Privacy leak

1. はじめに

スマートフォンの普及や，ユーザのプライバシーへの関心が高まる中，プライバシーデータの収集・使用に関する情報をユーザへ提供することが求められている．そこで，Androidアプリケーション（アプリ）の公式マーケットであるGoogle Play [1]に，ユーザに対し，アプリが収集，共有，保護するデータに関する情報を提供することを開発者に義務付ける「データセーフティセクション」が2022年に

[1] 岡山大学 Okayama University
[2] 名古屋工業大学 Nagoya Institute of Technology
[a)] poyr9jl9@s.okayama-u.ac.jp

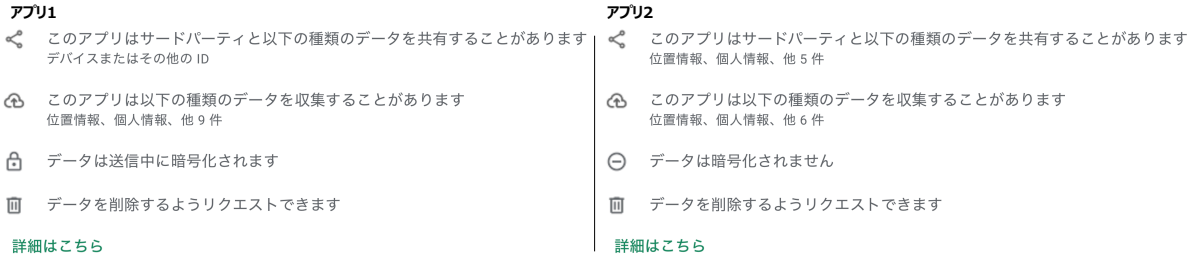

図 1 暗号化あり（アプリ 1）と暗号化なし（アプリ 2）を示すデータセーフティセクションの例
Fig. 1 Data safety section examples of an app with the encryption (app 1) and an app without the encryption (app 2).

導入された．2 つのアプリのデータセーフティセクションの例を，**図 1** に示す．上からデータの共有，収集，暗号化，削除リクエストに関する情報が掲載されている．データの送信中の暗号化（暗号化ラベル）に関して，暗号化される場合（アプリ 1）と，されない場合（アプリ 2）がある．

Arkalakis ら [2] は，データセーフティセクションで開示される情報と，アプリによる実際のデータの収集や共有の間には多くの矛盾が存在することを指摘した．現状では，データセーフティセクションは，Google Play [1] のユーザがより安心してアプリを利用する手助けになっておらず，また，開発者が十分にデータの取り扱いに注意できていない点が指摘された．一方で，Arkalakis らは暗号化ラベルに関しては調査していない．

暗号化ラベルに関するこれまでの調査では，暗号化されると記載しているにも関わらず，データを全く収集，共有しないと宣言するアプリや，通信に必要なパーミッションを求めないアプリが発見され，矛盾したラベルがユーザに混乱を招くと指摘された [3]．また，暗号化ラベルを含むデータセーフティセクションの内容と，プライバシーポリシーの内容との間の不一致が発見された [4]．以上のような問題が暗号化ラベルに関して調査されてきた一方で，アプリの実際の動作を考慮して暗号化ラベルの問題を調査した研究はなく，暗号化ラベルがアプリの動作を反映できているのかどうかに関する実態は明らかにされていない．

本稿は，暗号化ラベルとアプリの実際のデータ通信とを比較し，実態を調査する．国内の Google Play からランキング上位 6,180 アプリを収集し，Android 端末上で実行して通信を収集する．また，アプリの収集と同時に，各アプリの暗号化ラベルを抽出する．最後に，収集した通信と暗号化ラベルを照合し，ラベルに反して実際には暗号化しないアプリ（パターン 1）を検出する．加えて，ラベル上は暗号化しないとあり実際にも暗号化しないアプリ（パターン 2）を調査する．パターン 2 は，ラベルは一致しているためアプリ開発者にとっては問題ないが，逆にユーザにとってはラベル不一致（通信データ暗号化あり）の方が望ましい．パターン 2 を調べることで，ラベル上暗号化しないアプリをユーザは本当に避けるべきかどうかを明らかにする．

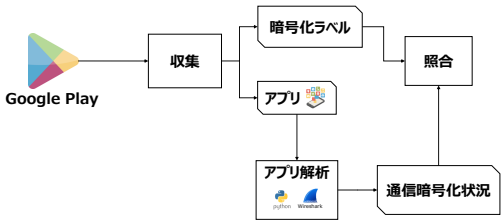

図 2 アプローチの全体像
Fig. 2 Overview of our approach.

本稿は調査の第 1 歩として，ユーザのアプリ操作を伴わずに発生する通信により収集されるデータ（これをアプリが自動的に収集するデータと呼ぶ）を対象とする．結果，パターン 1 を 11 アプリ，パターン 2 を 12 アプリ発見した．調査したアプリ数に対して小さいが，各アプリのインストール数は 1 万〜1 億であり，多くのユーザに影響が及んでいるため，軽視すべきではない．また，実験ではアプリを単に起動しただけであり，同意確認などの UI 操作は全く行っていないため，今後，UI 操作を加えることで，パターン 1 や 2 のアプリが増える可能性が高い．

本稿はアプリ開発者に対して，アプリのセキュリティを向上させるためにも，より積極的に暗号化を導入することを提言する．また，ユーザは，パターン 2 の結果から，暗号化しないとあるアプリは避けるべきであり，パターン 1 の結果から，暗号化するとあるアプリに対しても注意して，そのアプリが信頼できるかどうかを他の情報も参照して判断する必要がある．

以降，2 章で本稿のアプローチを紹介し，3 章で評価について報告し，4 章で制限と今後の課題を説明する．5 章で関連研究を挙げ，6 章でまとめを述べる．

2. アプローチ

アプローチの全体像を**図 2** に示す．まず，Google Play からアプリとデータセーフティセクションを収集し，データセーフティセクションから暗号化ラベルを抽出する（2.1 節）．そして，収集したアプリに対して動的解析を行い通信データを収集し，通信暗号化状況を調査する（2.2 節）．

表 1 調査対象とする，アプリが自動的に収集するプライバシーデータ
Table 1 Investigation target privacy data that apps automatically collect.

データカテゴリ	データの種類	説明	具体的なデータ	探索に使用した文字列	
				実際の値	その他
位置情報（1）	おおよその位置情報	3 平方キロメートル以上の地域でのユーザー またはデバイスの物理的な位置情報 （例: ユーザーがいる都市）.	IP アドレス 位置情報全般 県名	192.168.2*** - -	- location Okayama
	正確な位置情報	3 平方キロメートル未満の地域内にあるユーザー またはデバイスの物理的な位置情報.	緯度経度	34.6***, 133.9***	-
個人情報（2）	メールアドレス	ユーザーのメールアドレス.	Gmail アドレス	***@gmail.com	gmail
	住所	ユーザーの住所（送付先住所や自宅の住所など）.	住所全般	-	address
	電話番号	ユーザーの電話番号.	電話番号	080-9180-****	-
アプリのアクティビティ（3）	アプリインタラクション	ユーザーがアプリをどのように操作しているかに関する情報 （例: ページへのアクセス数，タップした項目）.	-	-	-
	アプリ内の検索履歴	ユーザーがアプリ内で検索した内容に関する情報.	-	-	-
	インストール済みのアプリ	ユーザーのデバイスにインストールされているアプリに関する情報.	-	-	-
	その他のユーザ作成コンテンツ	この一覧，または他のセクションに記載されていない， その他のユーザー作成コンテンツ （例: ユーザーに関する情報，自由形式の回答）.	-	-	-
	その他の操作	この一覧に記載されていない，その他のユーザアクティビティや アプリ内操作（ゲームプレイ，高評価，ダイアログでの選択など）.	-	-	-
アプリの情報，パフォーマンス（4）	クラッシュログ	アプリのクラッシュデータ （アプリがデバイスでクラッシュした回数， クラッシュに直接関係するその他の情報など）.	-	-	-
	診断	デバイスでのアプリのパフォーマンスに関する情報 （バッテリー駆動時間，読み込み時間，レイテンシ， フレームレート，技術的な診断など）.	-	-	-
	その他のアプリパフォーマンスデータ	ここに記載されていないその他のアプリのパフォーマンスデータ.	-	-	-
デバイスまたはその他の ID（5）	デバイスまたはその他の ID	各デバイス，ブラウザ，アプリに関連する識別子 （IMEI 番号，MAC アドレス，Widevine デバイス ID， Firebase インストール ID，広告 ID など）.	MAC アドレス SSID 広告 ID IMEI Android ID インストール ID インスタンス ID 識別子全般	e8:d5:2b:**:**:** iphone 076e10fb***など - - - - -	 - - imei android_id installation_id，install_id instance_id device_id

最後に，暗号化ラベルと照合する（2.3 節）.

2.1 アプリと暗号化ラベルの収集

Google Play [1] に掲載されるアプリとそのデータセーフティセクションを収集する．収集を効率化するために，Android 端末上の Google Play に対して，アプリのインストールからアンインストールまでの UI 操作を Python と adb [5] コマンドにより自動化した．アプリのインストール後に，端末上から別の計算機へそのアプリをコピーすることで保存する．また，データセーフティセクションを保存するために，“adb shell uiautomator dump” コマンドによりアプリの掲載ページを xml 形式で取得する．得られた xml ファイルから暗号化ラベルを抽出して保存する.

2.2 アプリの解析

アプリの通信暗号化状況を明らかにするために，アプリの通信を収集（ 2.2.1 項）し，通信が暗号化されているかどうかを判断（ 2.2.2 項）する．そして，暗号化されていない通信中から調査対象データを探索する（ 2.2.3 項）.

2.2.1 通信の収集

アプリの通信を収集するためのアプローチとして，mitmproxy [6] を検討した．しかし，mitmproxy をテストした結果，アプリの HTTPS 通信が遮断されてしまい，アプリが適切に通信を行うことができないことがわかった．mitmproxy により HTTPS 通信を中継するために，アプリを書き換える方法があるが，失敗する場合もある．本稿の調査では，HTTPS 通信の中身を覗く必要はなく，HTTPS 通信は遮断されずに適切に通過すれば問題ない．そこで，squid [7] プロキシを用いて通信を中継し，tcpdump [8] でアプリの通信パケットをキャプチャすることとした.

アプリを Android 端末上で実行して通信を収集するための，各アプリに対して実行する一連の手順を以下に示す.

(1) アプリの Android 端末へのインストール
(2) tcpdump の実行
(3) アプリの起動
(4) アプリ起動状態で一定時間待機
(5) スクリーンショットの撮影
(6) tcpdump の停止
(7) アプリのアンインストール
(8) 次のアプリへ移る前に一定時間待機

以上の処理を Python により自動化し，各アプリに対して実行する．なお，squid は常に起動している．アプリのインストール時に，adb コマンドのオプションによりアプリが要求するすべてのパーミッションを許可している．各アプリに対して，アプリ起動状態で 60 秒間待機する．待機中はアプリの操作は行わないこととした．アプリの UI に対して，Monkey [9] といった自動的にランダムな操作を注入するツールが存在するが，Monkey は別のアプリを立ち上げる可能性がある．アプリが複数同時に立ち上がった場合，複数のアプリの通信が混ざって記録され，正確な調査を実施できないため，本稿では，UI 操作はなしとした.

スクリーンショットの撮影は，アプリを操作しない状態

で表示される UI を記録するためである．ダイアログボックスやログイン画面が表示されるアプリの数を把握し，今後の研究で利用する（4 章）．また，次のアプリへ移る前に，2 秒間待機する設定で実験を行う．

キャプチャしたパケットの解析では，自動的に計測できるところは scapy [10] を用い，より詳細な分析は Wireshark [11] を用いる．

2.2.2 暗号化有無の判断

通信が暗号化されているかどうかの判断は，公式ガイドラインである Google Play Console [12] に従う．送信時のアプリデータを安全に暗号化するには，最適な業界基準に準拠する必要がある旨が記載されている．一般的な暗号化プロトコルとして，HTTPS や，TLS（Transport Layer Security）が挙げられている．よって，アプリの通信のあるパケットにおいて，HTTPS（TLS）が用いられていれば，そのパケットは暗号化されていると判断する．

一方で，HTTP が用いられている場合，通信データを覗くことは可能であるが，そのデータに対してアプリが独自に暗号化を実施している場合がある．独自に暗号化された通信データに含まれるデータを特定するためには，アプリ本体のコードを解析する必要があり，コストが高い．また，通信データに対して何らかのエンコーディングが行われている場合も同様である．本稿は，アプリの独自の暗号化やエンコーディングをスコープ外とし，次の 2.2.3 項で述べる調査対象データの具体的な値などの完全な一致に基づいて，暗号化されていないデータを検出する．

2.2.3 調査対象データ

調査対象とした，アプリが自動的に収集するプライバシーデータを **表 1** に示す．調査を簡単化するために，手動による UI 操作などは行わない方針とし，ユーザが明示的に（UI 操作により）提供するデータは対象外とする．データセーフティセクションの申告対象である 14 データカテゴリの内，5 カテゴリに属する 14 種類のデータを対象とした（1，2 列目）．それぞれのデータに関する公式の説明 [12] を 3 列目に示す．また，調査する具体的なデータを 4 列目に示している．

アプリを実行して収集した HTTP 通信のペイロードに対して，grep コマンドを用いて調査対象データを探索し，当該データの送信を検出する．探索に使用した文字列として，データの実際の値（5 列目）と，その他の文字列（6 列目）がある．IP アドレスや緯度経度など，実際の値を用意できるものは実際の値を使用する．また，例えば位置情報全般を検出するために，“location” という文字列を探索する．また，アプリのアクティビティとアプリの情報，パフォーマンスの 2 データカテゴリに対しては，実際の値の用意は困難であるため，代わりとしてファイルの送信を検知し，中身を手動で調査する．

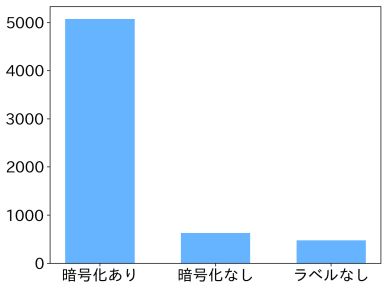

図 3 ラベル上の暗号化ありなしとラベルなしのアプリ数

Fig. 3 Number of apps with encryption labels stating yes and no and without encryption labels.

2.3 暗号化ラベルと実際の暗号化状況の照合

本稿は，次の 2 つのパターンを調査する．

- パターン 1: 暗号化すると申告しているが実際には暗号化していない．ラベルはアプリの実際の動作を反映しておらず，ユーザに間違った情報が提供されるため，問題である．
- パターン 2: 暗号化しないと申告しており実際にも暗号化していない．アプリ開発者にとっては正しい申告であり問題ないが，ユーザにとっては通信データが暗号化される方が望ましい．この調査により，ラベル上暗号化しないアプリをユーザは本当に避けるべきかどうかを明らかにする．

3. 評価

評価環境とアプリ（ 3.1 節），暗号化ラベルの収集結果（ 3.2 節），アプリ通信の収集結果（ 3.3 節），暗号化ラベルと実際の暗号化状況の照合結果（ 3.4 節）を順に報告する．

3.1 評価環境とアプリ

Android 端末として，Google Pixel の最新機種である Android 14 の Pixel 8a を 3 台使用した．また，使用するアプリセットの収集時期は 2024 年 2 月であり，Google Play [1] のアプリの 35 カテゴリすべてを対象として，それぞれのランキング掲載アプリを収集した．各ランキングで最大 200 程度のアプリが掲載されており，収集できたアプリの総数は，重複を除いて 6,180 個である．

3.2 暗号化ラベルの収集結果

2024 年 2 月時点のデータセーフティセクションから，6,180 個のアプリに対応づく暗号化ラベルを抽出した結果，暗号化されるとあるアプリは 5,076 個（82%），暗号化されないとあるアプリは 629 個（10%），暗号化ラベルがないアプリは 475 個（8%）であった（**図 3**）．

暗号化ラベルがないアプリは，開発者がラベルを設定していないためであると思われる．これらのアプリは，データ収集，共有ラベルも設定されていない，もしくは，収集しない，共有しないというラベルが設定されており，どち

表 2　収集した HTTP リクエストメソッドの数と説明
Table 2　Number and description of collected HTTP request methods.

メソッド	数	説明
GET	1,616	リソースの取得を目的とし，パラメータを URL に付与して送信
POST	242	リソースの作成を目的とし，データ本体をメッセージボディに格納し送信
PUT	3	リソースの更新
HEAD	52	リソースのヘッダ情報のみを取得
CONNECT	109,563	プロキシを介して双方向通信を開始
合計	111,476	

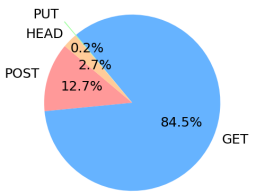

図 4　HTTP リクエストメソッドの割合（CONNECT 以外）
Fig. 4　Percentage of HTTP request methods (other than CONNECT).

らも収集，共有が行われないことを伝えている．この場合，暗号化に関わらず，プライバシーデータの収集があれば，ラベルと不一致であり問題である．しかし，今回調査した範囲では，暗号化ラベルなしかつプライバシーデータを送信するアプリは見つからなかった．

3.3　アプリ通信の収集結果

アプリを実行して通信を収集した結果，HTTP コネクションを 110,546 件，HTTPS コネクションを 108,134 件観測した．scapy を用いて HTTP のペイロードを抽出し，使用された HTTP リクエストメソッドを集計した結果を **表 2** の 1，2 列目に示す．5 つの HTTP リクエストメソッドが用いられ，合計で 111,476 個を収集できた．HTTP コネクション数よりも HTTP リクエストメソッド数の合計がやや多い理由は，1 つのコネクションに対して複数のリクエストが発生する場合があるためである．

各 HTTP リクエストメソッドの簡単な説明を 3 列目に示す．GET はサーバ上の特定の情報やリソースの取得を目的とし，パラメータを URL に含めて送信するメソッドである．観測数は 1,616 であり，比較的多い．POST はリソースの作成を目的とし，データ本体をメッセージボディに含めてサーバへ送信するメソッドである．得られた数は，242 であり，比較的少ない．PUT はリソースのデータを新しいデータに更新するメソッドである．収集数は，3 個と極めて少ない．HEAD は指定したリソースのヘッダ情報のみを取得するメソッドである．ヘッダ情報には，リソースのタイプ，サイズ，最終更新日時などの情報が含まれる．得られた数は，52 個と PUT に次いで少ない．最後の CONNECT は，プロキシを介して双方向通信を確立する場合に使用されるメソッドであるため，通常は発生せず，実験のために使用したプロキシが要因で発生したと考えられる．得られた数は，109,563 であり，大多数を占める．

CONNECT はプライバシーデータの送信には用いられず，実験環境が要因で発生したと考えられるため除外し，その他の 4 つの HTTP リクエストメソッドの割合を**図 4** に示す．GET（84.5%）と POST（12.7%）の割合が大きく，HEAD（2.7%）と PUT（0.2%）も少量存在する．

表 3　プライバシーデータ毎の検知アプリ数
Table 3　Detected apps per privacy data.

カテゴリ	データ	パターン 1	パターン 2	全体
1	IP アドレス	1	1	2
	位置情報全般	1	0	1
	県名	1	0	1
	緯度経度	2	4	6
3 & 4	操作・パフォーマンスデータ	1	0	1
5	SSID	1	0	1
	広告 ID	2	4	6
	Android ID	1	1	2
	インストール ID	1	0	1
	識別子全般	1	3	4
全体		11	12	23

3.4　暗号化ラベルと実際の暗号化状況の照合結果

収集した HTTP 通信のペイロードに対して 表 1 に示す文字列を探索した結果，4 データカテゴリの 10 データに関してパターン 1 及び 2 を発見した（**表 3**）．カテゴリ番号とカテゴリ名の対応は 表 1 に示している．パターン 1 を 11 個のアプリで，パターン 2 を 12 個のアプリで発見した．つまり，全体で 23 個のアプリで暗号化していない状態でのプライバシーデータの送信を検出した．パターン 1 の検出数の合計は 12 件であるが，1 つのアプリで SSID とインストール ID の送信を検出したため，アプリ数の合計は 11 個である．また，パターン 2 においても，検出数の合計は 13 件であるが，1 つのアプリで広告 ID と Android ID の送信を検出したため，アプリ数の合計は 12 個である．

パターン 1 の詳細を **表 4**，パターン 2 の詳細を **表 5** に示す．パターン 1 と 2 を合わせると，HTTP リクエストメソッドについては，GET が 12 個，POST が 12 個，PUT が 1 個であり，GET と POST が主なデータ送信手段である．検出されたアプリのカテゴリの傾向としては，天気アプリが 5 個と最も多い．このカテゴリのアプリをインストールする際には，特に注意が必要であると言える．その次に多いカテゴリとして，エンターテイメントが 3 個であった．

問題が検出された 23 個は，調査したアプリ数に対して小さいが，どれもランキング上位であり軽視すべきではない．

表 4　パターン 1 に該当するデータ通信を行ったアプリの詳細
Table 4　Details of apps that performed data transmission matched to the pattern 1.

プライバシーデータ	アプリ	HTTP メソッド	通信データ（一部抜粋）	アプリカテゴリ	インストール数
IP アドレス	1	POST	:“sdk”,“uip”:“192.168.2.20”	動画プレーヤー&エディタ	1 億
位置情報全般（“location”）	2	GET	/geoapi/elev/json?locations=139.770527,35.677836	地図&ナビ	100 万
県名（“Okayama”）	3	GET	weather/current?city=Okayama	健康&フィットネス	5 万
GPS 以外から取得された緯度経度	4	POST	loc=JP ｜ 33 ｜ 700-0931 ｜ 34.6401x133.9183	通信	1,000 万
	5	GET	data/2.5/weather?lat=34.68468468468468 &lon=133.92513377844628	天気	100 万
操作，パフォーマンスデータ	6	PUT	/app_log/2024-07-09/1720484073127580131%2APixel %208a%2A14%2A1720484074637%2A3.3.4%2AUS %2A32400%2A2%2A666%2A1.zip	ソーシャルネットワーク	10 万
SSID	7	POST	Android/Pixel 8a Network/%22iphone%22	ライフスタイル	10 万
広告 ID	8	POST	Advertising-Identifier : 5f3bff1b-875c-42a0-b129-1618062d47d5	エンターテイメント	5,000 万
	9	GET	&aid=5f3bff1b-875c-42a0-b129-1618062d47d5	エンターテイメント	10 万
Android ID	10	POST	android_id=3n95e54	イベント	5 万
インストール ID	7	POST	“a_install_id”:“684d9eda1e83d493401926e655b26f35”	ライフスタイル	10 万
識別子全般（“device_id”）	11	POST	device_id=eede2689-9013-411d-b132-743ac6443247	ニュース&雑誌	5 万

表 5　パターン 2 に該当するデータ通信を行ったアプリの詳細
Table 5　Details of apps that performed data transmission matched to the pattern 2.

プライバシーデータ	アプリ	HTTP メソッド	通信データ（一部抜粋）	アプリカテゴリ	インストール数
IP アドレス	12	POST	:“sdk”,“uip”:“192.168.2.18”	出産&育児	500 万
GPS から取得された緯度経度	13	GET	&version=3&client=android&build=2.42.43 &lang=en&lat=34.68998&lon=133.92345	天気	1,000 万
	14	GET	?lat=34.6899859&lon=133.9234313	天気	100 万
	15	GET	data/2.5/weather?lat=34.6899859&lon=133.9234313	天気	100 万
	16	GET	/reverse?point.lat=34.6899906&point.lon=133.9234367	天気	100 万
広告 ID	17	GET	&gaid=019bf53a-83cd-4433-a34c-ce6059405e64	ニュース&雑誌	1 万
	18	GET	?identify=076e10fb-e300-44e2-ac7f-de3f2fda657d	ソーシャルネットワーク	10 万
	19	POST	&adid=076e10fb-e300-44e2-ac7f-de3f2fda657d&	旅行&地域	500 万
	20	GET	mid=5f3bff1b-875c-42a0-b129-1618062d47d5	エンターテイメント	1 億
Android ID	18	GET	android_id=74c5a467d757b1e5	ソーシャルネットワーク	10 万
識別子全般（“device_id”）	21	POST	“device_id”:“2538ad41-aeb0-4c60-9bca-b17ea0765005”	出会い	5 万
	22	POST	device_id=4fc97dc710e4f6ea	フード&ドリンク	10 万
	23	POST	“device_id”:“b8c17d35-a1a7-4f40-b4ac-eec584078626”	出会い	10 万

インストール数は小さなものでも 1 万インストール（アプリ 17），大きなものでは 1 億インストール（アプリ 1，20）と多くのユーザに利用されており，影響が及んでいる．

本節ではこれ以降，表 4 と表 5 に示した検出結果の詳細について，プライバシーデータ毎の分析を述べる．

3.4.1　位置情報

IP アドレスに関しては，パターン 1 で 1 つ（アプリ 1），パターン 2 で 1 つ（アプリ 12）検出した．これらのアプリは同じ会社によって公開されており，IP アドレス送信先は同じで，ドメイン名にその会社名を含んでいる．公開元が同じであっても，アプリによって暗号化ありや暗号化なしのように異なる暗号化ラベルを付けてしまっており，きちんと管理されていない状況が明らかとなった．

位置情報全般を表す “location” を平文で送信するアプリをパターン 1 で 1 つ（アプリ 2）発見した．通信データ中の緯度経度の値は，東京を示しており，実験を実施した岡山県ではないため，問題は小さい．また，送信先は外部サービスである．アプリのカテゴリが地図&ナビであるため，アプリから位置情報を送信し，対応する地図情報を得ていると考えられる．また，緯度経度のような詳細な位置情報ではないが，県名を平文で送信するアプリをパターン 1 で 1 つ（アプリ 3）発見した．こちらも広いエリアを指すため，単体では個人の特定は困難であるが，他の情報と組み合わせることで個人の特定につながる恐れがある．

緯度経度については，パターン 1 で 2 つ（アプリ 4，5），パターン 2 で 4 つ（アプリ 13，14，15，16）の計 6 つ発見した．パターン 1 の 2 つで送信された緯度経度を調べた結果，アプリ 4 は岡山県岡山市北区神田町 1 丁目付近，アプリ 5 は岡山県岡山市北区学南町 3 丁目付近を指すものであった．これらは実験を行った場所からやや離れており，実験中に端末上の GPS から取得された位置情報とは考えづらいため，GPS 以外から取得されたものであると判断した．一方，パターン 2 の 4 個のアプリが送信した位置情報はすべて，実験を実施した岡山県岡山市北区津島中 3 丁目 1-1 の工学部 4 号館を指しており，GPS から取得されたものであると判断できる．このような位置情報を暗号化せずに送信することは，ユーザの正確な居場所の特定につながるため，アプリ開発者はより注意すべきである．アプリ

のカテゴリとしては，6 個中 5 個が天気であった．ユーザの位置情報に基づいて天気を取得・表示していると考えられ，位置情報の送信は自然な動作と思われるが，送信時の暗号化が行われない場合は問題であるため，ユーザ自身も注意を払ってアプリを選択する必要がある．

3.4.2 アプリのアクティビティとアプリの情報，パフォーマンス

パターン 1 に該当するアプリを 1 個（アプリ 6）発見した．アプリ 6 は，HTTP リクエストメソッドの PUT を用いて zip ファイルを送信していた．通信データ中から zip ファイルを抽出して展開すると，“CommonInfo.log” などの名前の付いたファイルが 8 個得られた．主に例外に関するログ，ファイルのダウンロードに関するログ，HTTP 通信に関するログ，アクティビティに関するログが含まれていた．これらは，ユーザのアプリ操作を反映した情報と考えられ，また，アプリのパフォーマンスデータでもある．

3.4.3 デバイスまたはその他の ID

SSID を平文で送信するアプリをパターン 1 について 1 つ（アプリ 7）発見した．SSID は，Wi-Fi 通信で利用されるネットワーク識別名である．本調査では，“iphone” を SSID として使用した．SSID は，個人を特定できる情報などを名前として使用される場合があるため，取扱には注意が必要なデータである．

広告 ID に関しては，パターン 1 で 2 つ（アプリ 8，9），パターン 2 で 4 つ（アプリ 17，18，19，20）検出した．アプリのカテゴリはエンターテイメントが 3 個と，ニュース&雑誌，ソーシャルネットワーク，旅行&地域が 1 個ずつであり，様々である．エンターテイメントのアプリのより細かいタイプとしては，2 個がストリーミング，1 個がファンコミュニティに関するアプリであった．広告 ID は，Android デバイスの識別に利用され，1 つのデバイスが 1 つの広告 ID を持っているため，同一のデバイスから送られたデータを受信側で紐づけることができる．本調査でも，異なる複数のアプリが広告 ID を収集していた．通信データから広告 ID を盗み取れる場合，自社アプリやライブラリがインストールされていない端末からも情報を集められる．こうして集められた情報に基づき，ユーザの興味などが推測され，広告などに利用される．ユーザによる広告のパーソナライゼーションの制御を尊重することが求められる中，広告 ID を送信中も保護することが重要である．

その他にも，Android ID に関して，パターン 1 で 1 つ（アプリ 10），パターン 2 で 1 つ（アプリ 18）発見した．Android ID は，デバイス固有の識別子であり，デバイスで初めて Google サービスにログインした際に自動生成され，記録される．デバイスを工場出荷状態に初期化するなど，特別な操作を行わない限り，再生成は行われない．また，インストール ID に関して，パターン 1 で 1 つ（アプリ 7）検出した．インストール ID は，アプリケーションがデバイスにインストールされた際に生成される一意の識別子である．さらに，識別子全般を表す “device_id” を探索した結果，平文で送信するアプリをパターン 1 で 1 つ（アプリ 11），パターン 2 で 3 つ（アプリ 21，22，23）発見した．これらの識別子も，広告 ID と同様に，ユーザの望まない追跡を防ぐために，通信中も保護されるべきである．

4. 制限と今後の課題

本稿は，アプリが自動的に収集するデータを対象としているが，実際に検索するデータタイプ名や具体的な値は著者らが準備した（表 1）．これらは不完全である可能性があり，アプリが自動的に収集するデータをすべてカバーしているわけではない．データタイプ名や具体的な値をさらに増やすことで，パターン 1 や 2 をより多く検出できる可能性がある．また，今後の課題として，ユーザが明示的に UI 操作により提供するデータを対象として再度調査を実施する．例えば，連絡帳に適当なアドレスを登録することで，連絡帳データをアプリに提供できると考えている．他にも，ファイルや写真，動画などの端末上のデータを充実させ，再度アプリの通信収集を行う．

実験では，アプリを自動的に起動し，その後何も操作を行わずに一定時間待機した後，アプリをアンインストールして次のアプリの解析へ進む．しかし，アプリが起動したとき，同意確認や認証の操作を求めるアプリが多くあり，アプリが本来行う通信を十分に収集できていないと考えられる．今後，UI 操作を手動で行うことで，パターン 1 や 2 のアプリが増える可能性が高い．

また，アプリの収集時期である 2024 年 2 月時点では，データセーフティセクションに暗号化ラベルが記載されていたが，2024 年 7 月時点では，暗号化ラベルがなくなったアプリが見つかった．また，パターン 1 からパターン 2 に切り替わったアプリも発見された．これらの場合に関して，詳細な調査が必要だと考えている．

パターン 2 の調査において，暗号化しないとあるのに実際には暗号化していたアプリは数えていない．構築した解析環境では，HTTPS のペイロードは検査できないためである．また，解析環境でアプリが動作する範囲には限界があるため，アプリが絶対にプライバシーデータを平文送信しないと断定することは難しい．

5. 関連研究

プライバシーラベルのコンセプトは 2009 年に提案 [13] され，ここ数年で Android や iOS のアプリストアへ導入された．過去の研究では，ラベルの不一致ばかりが問題とされていたが，今回のように暗号化の場合は，ラベルの一致がユーザにとって望ましくない場合もあり，本稿はパターン 2 として調査した．Android のプライバシーラベルに関して，Arkalakis らの研究 [2] では，送信データ暗号化と

データ削除要求以外の，データの収集，共有についてのラベルと実際のアプリの動作との不一致を調査している．また，Khandelwal らは，定量的手法と定性的手法を用いて，データセーフティセクションを包括的に分析している [3]．データを収集，共有しないと記載しているにも関わらず暗号化ありと記載しているアプリの割合や，通信パーミッションの要求がないにも関わらず暗号化ありと記載しているアプリの割合，無料アプリと有料アプリで暗号化ラベルを持つアプリの割合を調査した．主にラベル上の情報のみに注目しており，一方，本稿は実際の通信に注目している．

Ragab らは，AI チャットボットアプリによるデータの収集，共有について，データセーフティセクションとプライバシーポリシーとの間の不一致を調査した [14]．トラッキングライブラリの検出や，動的トラフィック分析によるサーバまたは第三者へ送信されるユーザデータの特定，また，そのデータを危険にさらす認証上の問題を調査した．Malki らは，女性のモバイルヘルスアプリのデータセーフティセクションとプライバシーポリシーを調査し，データの収集，共有に関する不一致に加え，暗号化ラベルとデータ削除要求に関する不一致も発見した [4]．

iOS のプライバシーラベルについて，iOS アプリ開発者がラベルを作成する様子の観察や，iPhone ユーザへのインタビューなどが実施された [15]．iOS のプライバシーラベルには暗号化ラベルがない．一方，Android のラベルはデータの安全性に重点を置いており，ユーザインタビューでは，暗号化ラベルなどのアプリのセキュリティ対策に関する情報を提供することを高く評価するユーザもいる．また，iOS のプライバシーラベルの改良のために，データの収集と使用を色分けしたグリッド形式による新たなデザインが提案されている [16]．

本稿はデータセーフティセクションの暗号化ラベルに注目しているが，EU に居住するユーザからデータを収集するすべての Android アプリが，GDPR [17] に準拠しているかに注目した研究 [18] では，Android アプリにおけるデータ保護状況の診断において，静的解析の必要性を述べ，その問題点と改善点を挙げている．

6. まとめ

本稿は，Android アプリの暗号化ラベルに関する実態を調査した．国内の 6,180 個のアプリを解析した結果，23 個のアプリで平文のデータの送信があり，その内 11 個はラベルと矛盾しており，12 個は，ラベル上は暗号化しないとあり，実際に暗号化していなかった．今後の課題として，アプリ解析時に，同意確認や認証の操作を追加してより多くの機能を動作させ，また，調査対象をユーザが明示的に提供するデータに拡大し，再度調査を行う．

謝辞 本研究は JSPS 科研費 JP20H05706，JP24K23863 の助成を受けた．

参考文献

[1] Google: Google Play, `https://play.google.com`. (Accessed 2024-06-17).

[2] Arkalakis, I., Diamantaris, M., Moustakas, S., Ioannidis, S., Polakis, J. and Ilia, P.: Abandon All Hope Ye Who Enter Here: A Dynamic, Longitudinal Investigation of Android's Data Safety, *USENIX Security Symposium*, (online), available from ⟨https://www.usenix.org/system/files/sec24fall-prepub-399-arkalakis.pdf⟩ (2024).

[3] Khandelwal, R., Nayak, A., Chung, P. and Fawaz, K.: Unpacking Privacy Labels: A Measurement and Developer Perspective on Google's Data Safety Section, *USENIX Security Symposium*, (online), available from ⟨https://www.usenix.org/system/files/usenixsecurity24-khandelwal.pdf⟩ (2024).

[4] Malki, L. M., Kaleva, I., Patel, D., Warner, M. and Abu-Salma, R.: Exploring Privacy Practices of Female mHealth Apps in a Post-Roe World, *Proceedings of the CHI Conference on Human Factors in Computing Systems* (2024).

[5] Google: adb, `https://developer.android.com/studio/command-line/adb`. (Accessed 2024-07-11).

[6] Maximilian Hils: mitmproxy, `https://github.com/mitmproxy/mitmproxy`. (Accessed 2024-06-17).

[7] squid-cache.org: squid, `https://www.squid-cache.org//`. (Accessed 2024-06-27).

[8] The Tcpdump team: tcpdump, `https://www.tcpdump.org/`. (Accessed 2024-06-27).

[9] Google Android Developers: Monkey, `tps://developer.android.com/studio/test/other-testing-tools/monkey/`. (Accessed 2024-06-27).

[10] Biondi, P.: scapy, `https://github.com/secdev/scapy`. (Accessed 2024-07-11).

[11] Combs, G.: Wireshark, `https://www.wireshark.org//`. (Accessed 2024-06-27).

[12] Google: Google Play Console, `https://support.google.com/googleplay/android-developer/answer/10787469?hl=ja`. (Accessed 2024-06-17).

[13] Kelley, P. G., Bresee, J., Cranor, L. F. and Reeder, R. W.: A "nutrition label" for privacy, *Proceedings of the Symposium on Usable Privacy and Security* (2009).

[14] Ragab, A., Mannan, M. and Youssef, A.: "Trust Me Over My Privacy Policy": Privacy Discrepancies in Romantic AI Chatbot Apps, *IEEE European Symposium on Security and Privacy Workshops*, pp. 484–495 (2024).

[15] Cranor, L. F.: Mobile-app privacy nutrition labels missing key ingredients for success, *Commun. ACM*, Vol. 65, No. 11, p. 26–28 (2022).

[16] Zhang, S., Klucinec, L., Norton, K., Sadeh, N. and Cranor, L. F.: Exploring Expandable-Grid Designs to Make iOS App Privacy Labels More Usable, *Symposium on Usable Privacy and Security*, pp. 139–157 (2024).

[17] The European parliament and the council of the European union.: General Data Protection Regulation (GDPR)., `https://eurlex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679`. (Accessed 2024-08-17).

[18] Khedkar, M. and Bodden, E.: Toward an Android Static Analysis Approach for Data Protection, *International Conference on Mobile Software Engineering and Systems*, pp. 65–68 (2024).