令和 2 年度 東邦大学理学部情報科学科 卒業研究

Android アプリケーションにおけるサードパーティー製 API での暗号技術利用動向の調査

学籍番号　5517097

山口千尋

金岡研究室

目 次

1　はじめに

概要文みてかくといいかも

2　前提知識

2.1　Android

Android とは、Googel 社が 2007 年に開発したスマートフォンやタブレット端末など携帯情報機器向けの Operating System、あるいは Android OS が搭載された端末を指す。主にスマートフォンの OS として広く普及しており、世界的に Apple 社の携帯機器向け iOS と市場を二分している。

2.2　Operating System

Operating System(以後 OS) とは、ソフトウェアの種類の 1 つで、機器の基本的な管理や制御のための機能や、多くのソフトウェアが共通して利用する基本的な機能などを実装したシステム全体を管理するソフトウェアのことである。

2.3　アプリケーション

アプリケーションとは、Application Software の略であり、ゲームや音楽プレイヤー、メールなど、スマートフォンの OS 上で動くソフトウェアのことを言う。

2.4　Android アプリケーション

Android アプリケーションとは、Android にインストール可能なアプリケーションである。主に、Java や Kotlin などのプログラミング言語で作成されている。Java プログラムをコンパイルして機械語に変換し、画像などのリソースと合わせてパッケージにすることでインストール可能である。

2.5　APK

APK とは、Android Application Package の略であり、Android 向けのを Android 端末にインストールできる形式にパッケージにしたもの、もしくはそのファイルのことである。入手方法は０００に後述する APK ストアからダウンロードする方法や、単体で公開されている APK ファイルをダウンロードする方法等が存在する。一般的に APK は“ .apk ”拡張子を持つ。ただし、.apk ファイル自体は zip 形式で圧縮されており、その中にはアプリケーションの動作に必要なさまざまなファイルが納められている。.apk ファイルに対して zip ファイルと同様の解凍処理を行い、得られるファイルのうち本研究に関連する項目を解説する。

- AndroidManifest.xml
  - Android アプリケーションの必要要件や、最初に起動されるアクティビティの記述がされている
  - zip の解凍処理により得られる AndroifManifest.xml はバイナリファイルの状態であるため、テキストエディタ等で内容を閲覧するためにはデコード処理が必要である
  - デコードされた AndroidManifest.xml の入手方法は後述する
- classes.dex

– Android アプリケーションのソースファイルを変換して Android で実行できるようにしまとめたファイルである
– 1 つの dex ファイルに含められるメソッドの数は 65,536 が上限であり、それ以上の数のメソッドが 1 つの Android アプリケーションに含まれる場合は、classes2.dex、classes3.dex …と複数ファイルに分割される

2.6 バイナリファイル

バイナリファイルとは、コンピュータプログラムによって読み書きや処理を行うことを前提に、文字コードの規約を用いずに任意のビット列によって構成されるデータを格納するものである。一方、テキストファイルは文字コードで規定された自然言語の文字と、表示制御のための少数の制御コードのみを含み、人間が容易に読み書きできる。テキストファイルはテキストエディタなどで表示して中にどんな文字が書かれているかを読むことができるが、バイナリファイルはその形式に対応したソフトウェア以外ではまったく内容を知ることはできない。ただし、バイナリエディタというソフトウェアによってどのようなバイト列が並んでいるかを見ることが可能である。

2.7 APK ストア

APK ストアとは、Android アプリケーション開発者の作成した Android アプリケーションの配信を代行するサービス、およびそれを行っている Web サイトのことである。Android の公式 APK ストアは、Android の公式 APK ストアである GooglePlay[ ]1 つのみであり、非公式の APK ストアは数多く存在する。

2.8 Android Developers

Android Developers とは、Android アプリケーション開発者向けの Android 公式 Web サイトのことである。Android の詳細やドキュメントが提供されている。公式ドキュメントといった場合 Android Developers を指す。

2.9 smali ファイル

smali とは、Android の Dalvik 仮想マシンで使用される開発者ファイルである。通常、Android アプリケーションに含まれている実行可能ファイルである。DEX(Dalvik Executable)（Dalvik 実行可能）ファイル（.apk ファイル）を逆コンパイルすることによって作成される。smali ファイルの取得には、Apktool[ ] を用いる方法と、baksmali[ ] を用いる方法がある。それぞれのツールの詳細は０００で説明する。

2.10 中間言語

中間言語とは、計算機が実行するコードを人間が理解できる形式で表現するための言語である。以下に本研究に関連する Dalvik バイトコードについての詳細な説明を述べる。

2.11 Dalvik バイトコード

Dalvik バイトコードとは、Android における中間言語である。Apktool 等を用いて APK より取得できる smali ファイルは、Dalvik バイトコードで記述されている。以下に、ソースコード 1、ソースコード 2 に Dalvik バイトコードの例と、対応するソースコードを示す。

Listing 1: 対応するソースコード

```
public int add(int a, int b) {
    int c = a + b;
    System.out.print(c);
    return c;
}
```

Listing 2: Dalvik バイトコードの例

```
# virtual methods
.method public add(II)I
    .locals 2
    .param p1, "a"    # I
    .param p2, "b"    # I

    .prologue
    .line 3
    add-int v0, p1, p2

    .line 4
    .local v0, "c":I
    sget-object v1, Ljava/lang/System;->out:Ljava/io/PrintStream;

    invoke-virtual {v1, v0}, Ljava/io/PrintStream;->print(I)V

    .line 5
    return v0
.end method
```

2.12 CUI

CUI とは、Character User Interface の略であり、コンピュータやソフトウェアが利用者に情報を提示したり操作を受け付けたりする方法の 1 つで、すべてのやり取りを文字によって行う方式のことである。

2.13 Linux

Linux とは、Windows や macOS といった OS の 1 つである。Linux ディストリビューションの 1 つに Ubuntu がある。Ubuntu は、CUI でファイル操作が可能である点や、シェルスプリクトを利用して smali ファイルの解析を行うために本研究で利用した。

表 1: 正規表現における基本的なメタ文字の一覧

.	任意の 1 文字
∗	直前のパターンの 0 回以上繰り返し (最長一致)
+	直前のパターンの 1 回以上繰り返し (最長一致)
?	直前のパターンの 0〜1 回繰り返し (最長一致)

2.14 シェル

シェルとは、「オペレーティングシステムと対話するためのインターフェイス」であり、コマンドなどを制御する「環境」のことである。シェルがあることでコマンドを受付、OS との対話が可能である。CUI 環境においてシェルは最も身近なインターフェイスである。

2.15 UNIX コマンド

UNIX コマンドとは、Linux OS 等の UNIX マシンにおいて CUI 上からコンピュータを操作するために使用するコマンドを指す。ファイルのコピーを行う cp、ファイルの内容を表示する cat、ディレクトリの内容を表示する ls などが存在する。

2.16 シェルスプリクト

シェルスプリクトとは、OS を操作するためのシェル上で実行できる簡易なプログラム言語（スプリクト言語）のことを言う。また、スプリクト言語によって書かれた、複数の OS コマンドや制御文などを組み合わせたプログラムを指す。sh コマンドの引数としてシェルスクリプトのファイルを与えて実行すると、ファイルに記述された UNIX コマンドが上から順に実行される。以下のシェルスクリプトを実行すると、a.txt が b.txt にコピーされ、a.txt の末尾に “ hoge ” の文字列が追加される。

Listing 3: シェルスクリプトの例

```
cp a.txt b.txt
echo "hoge" > a.txt
```

2.17 正規表現

正規表現とは，ある文字列の規則を表現する方法である。正規表現ではメタ文字と呼ばれる特別な意味を持つ文字や記号が存在する。基本的なメタ文字を表 1 に示す。ある文字列の中から通常の文字とメタ文字によって作られた特定の規則に当てはまる文字列を検索するときに利用される。正規表現の例を表 2 に示す。

表 2: 正規表現の例

正規表現の例	正規表現の例の意味	マッチする例
.	任意の 1 文字	a
and∗roid	an と d の 0 回以上の繰り返しと roid からなる文字列	anroid
and+roid	an と d の 1 回以上の繰り返しと roid からなる文字列	andddddroid
and?roid	an と d の 0 回〜1 回の繰り返しと roid からなる文字列	android

2.18 暗号技術

2.18.1 MD5

MD5 とは、Message Digest algorithm 5 の略であり、ハッシュ値を計算するためのハッシュ関数の 1 つである。RSA 暗号の開発者の一人、ロン・リベスト氏らによって開発された。IPsec や、POP before SMTP など、さまざまなセキュリティプロトコルで使われている一方、最近になって脆弱性も指摘されている。ハッシュ関数により生成された値は「ハッシュ値」(hash value) と呼ばれる。

2.18.2 SHA-1

SHA-1 とは、アメリカ国家安全保障局が考案し、1995 年から米国政府の標準として使用されているハッシュ関数である。任意のデータから 160bit のハッシュ値を生成する。2017 年、Google が SHA-1 でハッシュ値が衝突する事例 [] を発見したため、より安全なハッシュ関数を使用することが推奨されている。

2.18.3 SHA-2

SHA-2 とは、SHA-1 を改良したハッシュ関数である。このハッシュ関数は、バリエーション豊富であり以下を総称して SHA-2 と呼ばれている。

- SHA-224（ハッシュ値：224bit）
- SHA-256（ハッシュ値：256bit）
- SHA-384（ハッシュ値：384bit）
- SHA-512（ハッシュ値：512bit）
- SHA-512/224（ハッシュ値：224bit）
- SHA-512/256（ハッシュ値：256bit）

ベースは SHA-256 と SHA-512 である。SHA-224 は SHA-256 で出力されたハッシュ値を 224bit に切り詰めたものであり、SHA-384 は SHA-512 で出力されたハッシュ値を 384bit に切り詰めたものである。SHA-512/224 と SHA512/256 についても SHA-512 で出力されたハッシュ値を 224bit、256bit に切り詰めたものである。大きな違いとしては、SHA-256 は 32bitCPU、SHA-512 は 64bitCPU に最適化されている点がある。ハッシュ長が長い方がセキュリティ的な強度が高いが、負荷が高く

なる。ただし、現状 SHA-256 でも必要十分な強度となっているため、一般的には SHA-256 が利用されている。

2.19 API

API とは、Application Programming Interface の略であり、あるコンピュータプログラム（ソフトウェア）の機能や管理するデータなどを、外部の他のプログラムから呼び出して利用するための手順やデータ形式などを定めた規約である。これは、3 種類に分けられる。詳細は、０００で説明する。

2.20 API ドキュメント

API ドキュメントとは、API による開発方法やクラス内のメソッドの使用方法を解説した説明書である。API リファレンスとも呼ばれる。

3　関連研究

本研究における関連研究を紹介する。

3.1　河合らの調査

河合による調査は、Android アプリケーションを調査対象とし、Android アプリケーションの暗号技術利用に関する現状を明らかにするために暗号で用いられるメソッド名や特徴のある用語によるフィルタリングアルゴリズムが指定可能な代表的箇所の抽出や API の利用傾向分析をしていた。ていた。しかし、河合の研究では公式 API のみの調査しか行われていない。

3.2　Y に関連した研究

あああ

- まずは
- この章のなかで書くことを
- 箇条書きで書き出してみる
- ことから始めましょう

4　調査ターゲット (準備)

・APK の取得方法　・smali ファイル化方法

・APK に使われている暗号技術　APK に使われている暗号技術は 3 種類ある

－公式 API 公式 API とは、Android の開発者向け公式 Web サイトである Android Developers の API リファレンスに記載されている API である。

－サードパーティ製 API サードパーティー製 API とは、サードパーティが提供する API のことである。サードパーティとは、特定のハードウェア、OS、ソフトウェア、あるいはサービスなどを対象として、それに対応する製品を販売、提供している組織や企業のことを指す。Google 社の tink や Facebook 社の Concealn がある。

－独自実装等の API API 開発者が既存の API を利用せずに独自に実装した API や、先述 2 つに含まれないものを独自実装等の API と本論文では呼ぶこととする。

API の分析　－公式 API の分析　河合さんがやった　－サードパーティ製 API の分析　今回は、Google 社の API、Tink に対して研究を進めていく　－独自実装等の API の分析　残課題

・Tink の分析 Google の Tink には 4 つのプリミティブがある　-プリミティブ　関連データを備えた認証付き暗号 (プリミティブ: AEAD)　メッセージ認証コード (プリミティブ: MAC)　ディジタル署名 (プリミティブ: PublicKeySign と PublicKeyVerify)　ハイブリッド暗号化 (プリミティブ: HybridEncrypt と HybridDecrypt)

・API の取得方法

5　調査手法

あああああ
あああああ

- 調査方法　－シェル作成内容の解説　　コマンド解説

6　調査結果

- APK で暗号、セキュリティに関する API がどれぐらい使われているのか（割合）　- アルゴリズムを指定して取得できた中で最も使われているもの（回数）　- 今回の調査結果と河合さんの結果との比較

- まずは
- この章のなかで書くことを
- 箇条書きで書き出してみる
- ことから始めましょう

7　今後の課題

- Google 以外のサードパーティー製 API の調査　- 独自実装等の API の調査

8　まとめ

まとめえええええええええええええええええええ

参考文献

[1] だれだれ, ”文献 1”, 年度

[2] だれだれ, ”文献 2”, 年度

[3] だれだれ, ”文献 3”, 年度

[4] だれだれ, ”文献 4”, 年度

[5] だれだれ, ”文献 5”, 年度

[6] だれだれ, ”文献 6”, 年度

[7] だれだれ, ”文献 7”, 年度

[8] だれだれ, ”文献 8”, 年度

[9] だれだれ, ”文献 9”, 年度

[10] だれだれ, ”文献 10”, 年度