

令和3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	4
2	前提知識	5
2.1	Android	5
2.2	Operating System	5
2.3	Android アプリケーション	5
2.4	APK	5
2.5	バイナリファイル	6
2.6	APK ストア	6
2.7	Android Developers	6
2.8	smali ファイル	6
2.9	中間言語	6
2.10	Dalvik バイトコード	6
2.11	Liux	7
2.12	Ubuntu	7
2.13	シェル	7
2.14	UNIX コマンド	7
2.15	シェルスクリプト	8
2.16	php	8
2.17	Python	8
2.18	正規表現	8
2.19	暗号技術	9
2.19.1	MD5	9
2.19.2	SHA-1	9
2.19.3	SHA-2	9
2.20	API	10
2.20.1	公式 API	10
2.20.2	サードパーティー製 API	10
2.20.3	独自実装等の API	10
2.21	API ドキュメント	10
3	関連研究	11
3.1	河合らの調査	11
3.2	Y に関連した研究	11
4	提案手法のメインな部分	12
5	提案手法の試作みたいなのを書く部分	13
6	試作を用いて評価	14
7	残課題	15



## 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、Google 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 のことである。また Android OS が搭載された端末のことである。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 2 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 Android アプリケーション

Android アプリケーションとは、Java という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 apk というパッケージにすることで Android にインストールできるアプリである。

### 2.4 APK

APK とは、Android Application Package の略であり、Android 向けのアプリケーションを 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” という拡張子を持つ。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 AndroidManifest.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

## 2.5 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はできる。

## 2.6 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。

## 2.7 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。

## 2.8 smali ファイル

smali とは、Android の Dalvik 仮想マシンで使用されるアセンブリ言語 Smali で書かれた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。

## 2.9 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。

## 2.10 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
```

## 2.11 Liux

Linux とは、OS の一種で、パソコンを動かすのに必要な基本ソフトウェアの 1 つである。

## 2.12 Ubuntu

Ubuntu とは、Linux 系の OS のことである。

## 2.13 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 との対話ができるようになる。CUI(Character User Interface) 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。

## 2.14 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 UNIX コマンドと呼ぶ。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。

## 2.15 シェルスクリプト

シェルスクリプトとは、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（スクリプト言語）。また、スクリプ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。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” という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
```

## 2.16 php

php とは、スクリプト言語の 1 つであり、php コマンドにより実行することができる。ソースコード 3 と同じ動作をする php プログラムを以下に示す。

Listing 4: php プログラムの例

```
<?php
copy('a.txt', 'b.txt');
$fp = fopen('a.txt', 'w');
fwrite($fp, 'hoge');
fclose($fp);
?>
```

## 2.17 Python

Python とは、スクリプト言語の 1 つであり、python コマンドによって実行することができる。ソースコード 3 と同じ動作をする Python プログラムを以下に示す。

Listing 5: Python の例

```
# -*- coding: utf-8 -*-
import shutil
shutil.copy('a.txt', 'b.txt')
with open('b.txt', 'a') as f:
    f.write('hoge')
```

## 2.18 正規表現

正規表現とは、ある特定のパターンをもつ文字列を指定する表記法の 1 つ。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利用される。正規表現の例を表 2 に示す。



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)

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	anddddddroid
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

## 2.19 暗号技術

### 2.19.1 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ひとつで、RSA 暗号の開発者のひとり、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。

### 2.19.2 SHA-1

SHA-1 (シャーフン) とは、Secure Hash Algorithm 1 の略であり、入力データを一定の手順で計算を行い、入力値のデータの長さに関わらず決まった長さの文字列を出力するハッシュ関数の 1 つ。生成された値は「ハッシュ値」(hash value) と呼ばれる。SHA-1 は NSA (米国家安全保障局) が考案し、1995 年に NIST (米国標準技術局) によって連邦情報処理標準の 1 つ (FIPS 180-1) として標準化された。2005 年頃から効率的に攻撃する手法がいくつか発見され十分な安全性が保たれなくなったため、近年では 2001 年に制定された後継の SHA-2 規格への移行が進んでいる。

### 2.19.3 SHA-2

SHA-2 とは、ハッシュ関数の計算手順 (アルゴリズム) を定義しており、どんな長さのデータからも常に同じ長さのハッシュ値を生成する。同じ原文からは必ず同じ値が得られる一方、少しでも異なる原文からはまったく違う値が得られる。データの伝送や複製を行なう際に、入力側と出力側でハッシュ値を求め一致すれば、途中で改竄や欠落などが起こっていないことを確認することができる。また、暗号や認証、デジタル署名などの要素技術として様々な場面で利用されている。

## 2.20 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。

### 2.20.1 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。

### 2.20.2 サードパーティー製 API

サードパーティーとは、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（プラットフォーム上で動作する、もしくは互換性のある）製品を販売・提供しているという意味である。企業が提供しているものや、開発者が提供しているものがあるサードパーティー製 API という。Google 社の tink(ティンク) や Facebook 社の Concealn(コンシール) がある。

### 2.20.3 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 とする。

## 2.21 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。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 提案手法のメインな部分

## 5 提案手法の試作みたいなのを書く部分

あああああ

あああああ

- 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

## 6 試作を用いて評価

あああああ

あああああ

- 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

## 7 残課題

サードパーティー製 API

## 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