卒業論文 2012年度（平成24年度）

スマートフォンアプリケーションの通信内容検証プラットフォームの設計と実装

慶應義塾大学 環境情報学部

氏名：小松 真

担当教員

慶應義塾大学 環境情報学部

村井 純

徳田 英幸

楠本 博之

中村 修

高汐 一紀

Rodney D. Van Meter III

植原 啓介

三次 仁

中澤 仁

武田 圭史

平成25 年1 月22 日

卒業論文要旨 - 2012 年度 (平成 24 年度)

スマートフォンアプリケーションの通信内容検証プラットフォームの設計と実装

近年，スマートフォンの急速な普及に伴い，膨大な数のアプリケーションが市場に公開されている．それらの中には，個人のプライバシを侵害するアプリケーションが多く存在し，スマートフォン利用の際の深刻な問題となっている．しかし，ユーザインターフェースからは，スマートフォンアプリケーションがプライバシを侵害しているか判断することが難しく，また，通信を解析して検査するには，ネットワークの知識と手間が必要である．更に，既存の研究では，プライバシを侵害するアプリケーションを総合的に検査できないといった欠点があった．

そこで，本論文では，スマートフォンアプリケーションがプライバシを侵害する動作をしていないか，ネットワークの知識がなくても，容易に検査できる環境を実現することを目標とした．その実現のために，アプリケーションの通信内容を検証し，自動でプライバシ侵害度を判定することによって，プライバシ面での不正を検知する手法を提案した．そして，提案手法の有効性を実証するために，提案手法に基づいたシステムを実装し，そのシステムを実際に市場のアプリケーションに対して使用することによって，検証を試みた．また，既存のスマートフォンアプリケーションの検査方法と比較し，本提案手法の優位性を示した．

その結果，本論文が提案する手法を用いることによって，アプリケーションのプライバシ面でのリスクを容易に把握することが可能になり，安心してスマートフォンを使用出来る環境の実現が期待される．

キーワード:
1. スマートフォン, 2. プライバシ, 3. ネットワーク解析, 4. 情報セキュリティ,

慶應義塾大学 環境情報学部

小松 真

Abstract of Bachelor's Thesis - Academic Year 2012

Design and Implementation of Network Inspection Platform for Smartphone Applications

In recent year, as smartphone become more widespread, more smartphone applications are available on the market. However, we are concerned with some smartphone applications which invade your personal privacy. It is difficult to figure out whether smartphone applications invade your privacy from their user interface. Moreover, you should have network knowleage to inspect communication of smartphone applications.

Therefore, the purpose of this study is creating a platform which inspect smartphone applications without technical knowleage. In order to achieve this, I suggest a method to analyze communication of smartphone applications and automatically judge their risk level on privacy. To prove the effeectiveness of this method, I implemented the system and did test for smartphone applications on the marcket by using the system.

As a result, it was shown that you will be able to inspect smartphone applications easily by using the method. It is expected that from the result of this thesis, you can use smartphone more safely.

Keywords :
1. Smartphone, 2. Privacy, 3. Network Analysis, 4. Internet Security,

Keio University, Faculty of Environmental and Information Stadies

Komatsu Makoto

目 次

図 目 次

表 目 次

第1章　序論

本章では，背景としてスマートフォンの現状について述べ，その後現状のスマートフォンアプリケーションの検査の問題点について述べる．そして，挙げた問題点を踏まえ，本研究の概要と目的について述べ，最後に本論文の構成を記す．

1.1　スマートフォンの現状

近年，小型端末の高機能化に伴い，パーソナルコンピュータの機能を備えた携帯電話であるスマートフォンが急速に普及している．シード・プランニングの調査 [1] によると，2011 年の世界のスマートフォン普及台数は 8 億 7200 万に及び，人口普及率は 12 %に達した．特に，オペレーティングシステムとして，Google の Android[2] と Apple の iOS[3] が広く市場を獲得しており，サードパーティによって，膨大な数のアプリケーションが公開されている．アプリケーショントラッキング企業の Appsfire の調査 [4] によると，App Store[5] には 100 万件を超える iOS アプリケーションが登録されており，また，BusinessWeek 誌の発表 [6] によると，Google Play[7] には 70 万件を超える Android アプリケーションが登録されている．

しかしながら，それらの中には，ユーザのプライバシを侵害する不正なアプリケーションが含まれており，スマートフォンの利用において深刻な問題となっている．

また，スマートフォンは，常時持ち歩いて使用され，GPS などの各種センサやインターネットの利用頻度が高いという特性から，プライバシに関わる情報を端末内に保持していることが多い．したがって，不正なアプリケーションをインストールした場合，連絡先情報や行動履歴などの個人情報が盗まれ，不正利用やなりすましなどの被害に遭う危険性が高い．

1.2　スマートフォンアプリケーションの検査の問題点

上記の背景に加え，スマートフォンアプリケーションは，どういった情報をどこのサーバーに送信しているのか，インターフェースからでは判断するのが難しいという問題がある．そのため，スマートフォンアプリケーションの通信を検証するためには，tcpdump[8] や Wireshark[9] などのツールを使って，ネットワークを解析する必要がある．しかし，これらのツールを使用するには，ネットワークに精通していなければならず，直感的に理解できないといった問題がある．また，スマートフォンアプリケーションの通信の検査に特化していないため，時間や手間がかかるといった欠点がある．

1.3 本研究の目的

本研究の目的は，スマートフォンアプリケーションの通信を自動的に解析することによって，アプリケーションによるプライバシ侵害を検知し，リスクレベルをユーザに分かりやすく表示するプラットフォームの構築である．これにより，ユーザは使用するアプリケーションのプライバシ面での危険度を容易に把握でき，より安心してスマートフォンを使用できるようになることが期待される．

1.4 本論文の構成

本論文は全 7 章から構成される．第 2 章では，ユーザのプライバシを侵害するアプリケーションを挙げ，既存の対策方法とその問題点を挙げる．第 3 章では，スマートフォンアプリケーションの検査に関する既存の関連研究を述べ，第 4 章では，第 2 章を踏まえて，通信内容の検証による検査手法を提案する．第 5 章では，第 4 章で述べた手法の設計と実装を行い，第 6 章では，第 5 章で実装したシステムに対して，評価と考察を行う．最後に第 7 章で本論文の結論と，今後の展望を述べる．

第2章　スマートフォンにおけるプライバシ侵害の現状と対策

本章では，最初に，スマートフォンにおけるプライバシの扱いについて述べる．その後，プライバシ面で不正な通信を行うスマートフォンアプリケーションを3つに分類するとともに，それによって引き起こされるプライバシ上の脅威について示す．そして，ベンダによるプライバシ侵害への対策や，既存のネットワーク検査ツールを挙げ，その問題点を述べる．

2.1　スマートフォンにおけるプライバシ

本節では，現状におけるスマートフォンの個人情報の扱いと，その問題点について述べる．

2.1.1　プライバシポリシー

アプリケーションにおける，個人情報の取り扱いの際の基準・方針を定めたものとして，プライバシポリシーがある．

しかし，プライバシポリシーは，文章が長く内容が難解であるため，ユーザによって読み飛ばされることが多い．また，収集された個人情報が何に使われるのか明記されていなかったり，虚偽の説明が記載されているものや，プライバシポリシーを提示しないアプリケーションも存在し，ユーザがアプリケーションを安全に使用することへの妨げとなっている．

プライバシポリシーの表示例を図2.1に示す．

2.1.2　個体識別番号

スマートフォンデバイスを一意に識別する番号として，UDID(Unique Device Identifier)やIMEI(International Mobile Equipment Identity)，MACアドレス(Media Access Control address)などが挙げられる．アプリケーションデベロッパや広告業者は，これらの個体識別番号を用いて，ユーザアカウントの管理や，アクセス解析を行う．しかし，個体識別番号は，ユーザによる変更や削除が難しいことが多く，他の個人情報と紐付けられることによって，プライバシを侵害されたり，ユーザの行動を追跡される恐れがある．

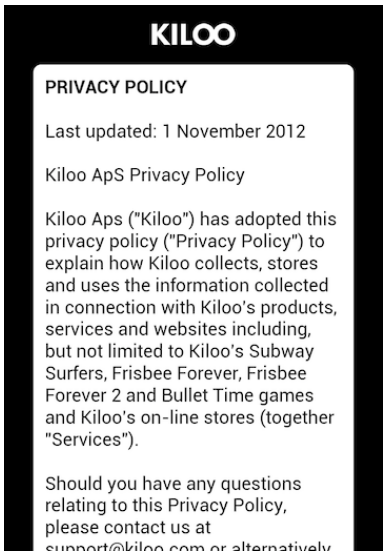

図 2.1: プライバシポリシーの表示例

また，個別識別子は，MD5 や SHA-1 などによってハッシュ化されていても，ハッシュ関数が公知であれば，結局，個別識別子は一意でありえるので，プライバシ侵害の恐れは残る．

2.2 不正な通信を行うアプリケーションの種類

スマートフォンの普及によって，ユーザに利益がもたらされていると同時に，不正な通信を行うスマートフォンアプリケーションによって，ユーザのプライバシが脅かされている可能性がある．そこで本節では，プライバシ面で不正な通信を行うスマートフォンアプリケーションを 3 つに分類するとともに，事例を挙げながらその特徴を述べる．

2.2.1 平文で認証情報を送信するアプリケーション

本項では，パスワードやセッション情報などの認証情報を，SSL/TLS などで暗号化せずに，平文で外部に送信するスマートフォンアプリケーションについて述べる．

ウルム大学の研究 [10] によれば，バージョン 2.3.3 以下の Android において，認証トークンを平文で送信する脆弱性があったことが明らかにされた．これは，Google Calendar や Google Contact など，ClientLogin という認証プロトコルを使用しているアプリケーションにおいて，認証用トークンが暗号化されずにやり取りされている問題に起因する．

一方，iOS アプリケーションにおいても，人気写真共有アプリケーションの Instagram[11] や，位置情報に基づいた SNS アプリケーションの Foursquare[12] が，平文でパスワードを送信していたことが明らかにされている．

このように，認証情報を平文で送信しているアプリケーションを使用すると，Evil Twin Attackなどによって，悪意のある第三者に通信を盗聴される恐れや，セッションハイジャックの被害に恐れがある．Evil Twin Attackとセッションハイジャックについては，2.3節で述べる．

また，独立行政法人情報処理推進機構（IPA）技術本部 セキュリティセンターが2011年12月20日に発表した調査[13]によると，インターネットユーザの8割以上がIDやパスワードといった認証情報を複数サイトで併用していることが明らかになった．したがって，プライバシに関わる情報を扱わないアプリケーションであっても，平文で認証情報を扱っていたことによって，通信を盗聴され，パスワードが流出することによって，連鎖的に他のサービスでもなりすまし被害が拡大する恐れがある．

2.2.2 ユーザに無断で個人情報を送信するアプリケーション

本項では，個人情報をユーザに無断で，開発者のサーバや広告会社に送信するスマートフォンアプリケーションについて述べる．

KDDI研究所の研究[14]によると，Androidの無料人気アプリケーション400件の5分間の挙動を調査した結果，45%のアプリが利用者に無断で，連絡先情報や個体識別番号などの個人情報を，アプリケーションデベロッパや，広告会社に送信していた．加えて，90%がインターネットのアクセス，58%が電話番号やSIM情報の読み取り，28%が位置情報へのアクセスの許可を求めていたことが明らかになった．

また，iOSにおいても，THE STREET WALL JOURNALの調査[15]によって，複数のアプリケーションが，個人情報をユーザの同意なく収集し，広告ネットワーク運営者などの第三者に提供していたことが明らかにされている．

このように，ユーザの個人情報が収集，利用されることによって，ユーザに最適なターゲット広告を提示されるようになる反面，個人情報が外部に送信・蓄積され悪用される恐れがある．

2.2.3 SSL/TLSの処理に不備があるアプリケーション

本項では，SSL/TLSの処理に不備があるスマートフォンアプリケーションについて述べる．

ライプニッツ大学とマールブルク大学の研究[16]によると，Google Playの人気の高い13500件の無料Androidアプリケーションのうち，1074本のアプリケーションに，SSL/TLSの処理に不備があることが述べられている．これは，Google Play人気上位100位までの無料アプリケーションに限定すると，41件に上記の脆弱性が存在した．これらのアプリケーションは，SSL/TLSの認証時に，あらゆる証明書，またはあらゆるホスト名を受け入れてしまい，偽の証明書であっても信用してしまう．

また，iOSにおいても，人気SNSアプリケーションのPath[17]が，SSL/TLSの証明書エラーを無視する実装になっていた[18]．

このようなSSL/TLSの処理に不備があるアプリケーションを使用した場合，悪意のある第三者によって中間者攻撃を仕掛けられ，個人情報やパスワードなどのアカウント情報を不正に取得されたり，悪意のあるソフトウェアを仕込まれる恐れがある．

中間者攻撃については，2.3節で述べる．

2.3 脅威の事例

本節では，プライバシ面で不正な通信を行うスマートフォンアプリケーションを使用した場合に，起こりうる脅威について述べる．

Evil Twin Attack

Evil Twin Attack（悪魔の双子攻撃）とは，悪意のある第三者が，無線ネットワーク上で公衆無線LANなどのアクセスポイントを偽装し，ユーザを騙す攻撃手法である．スマートフォンの多くは，一度接続したネットワークに二度目以降は自動で接続する設定になっている．したがって，Evil Twin Attackによって公衆無線LANと同名の偽アクセスポイントが仕掛けられていた場合，自動的にそのアクセスポイントに接続してしまい，悪意のある第三者によって，通信内容を盗聴されたり，改竄されてしまう恐れがある．

Evil Twin Attackの概要を図2.2に示す．

セッションハイジャック

セッションハイジャックとは，悪意のある第三者が，セッションを管理するセッションIDを推測や盗聴するなどして取得し，正規のユーザになりすまして通信を乗っ取る攻撃手法である．セッションIDが類推可能な規則的な文字列であったり，セキュアでない通信を行なっていた場合に被害に遭うことが多く，個人情報が盗まれたり，なりすましの危険性がある．

中間者攻撃

中間者攻撃（Man in the middle attack）とは，悪意のある第三者がクライアントとサーバとの通信の間に介入し，クライアントとサーバが交換する公開情報を自分のものとすり替えることにより，正規の相互認証が行われているようにして，セキュリティを破る攻撃手法である．多くの暗号プロトコルは，特に中間者攻撃を防ぐためのエンドポイント認証を含んでいるが，クライアント側の認証に問題があるなどした場合，中間者攻撃が成功してしまい，通信の盗聴や，通信内容を改竄されてしまう恐れがある．

図 2.2: Evil Twin Attack の概要

2.4 プライバシ侵害への対策

2.4.1 Android におけるプライバシ侵害への対策

Bouncer

Bouncer とは，Google によってつくられた，不正アプリケーション検出システムである [19]．Google Play に新規に公開されるアプリケーションと既に公開されているアプリケーション，および開発者アカウントを自動的にスキャンし，規約に違反するものを検出，削除する．Google によると，Google Play から削除された不正なアプリケーションのうち，40 %は Bouncer の検出によって削除されている．

しかし，セキュリティ企業 Duo Security の Jon Oberheide とセキュリティ研究者の Charlie Miller によって，Bouncer を回避して，不正なアプリケーションを公開する方法が発表された [20]．このように，検出システムは完全でなく，アプリケーションにまつわるセキュリティへの不安は依然として残っているのが現状である．

また，Bouncer の検査対象は Google Play のアプリケーションのみであり，Google Play 以外のマーケットで不正なアプリケーションが配布される危険は防げないといった問題がある．

パーミッションの表示

Android プラットフォームには，位置情報や連絡先情報など，アプリケーションが利用するデータに対するアクセスする権限（パーミッション）が明示される仕組みがある [21]. これによりユーザは，インストール時にパーミッションを確認することで，プライバシを侵害するアプリケーションであるか，ある程度の指標を得ることができる．図 2.3 に，Android アプリケーションにおけるパーミッション確認画面の例を示す.

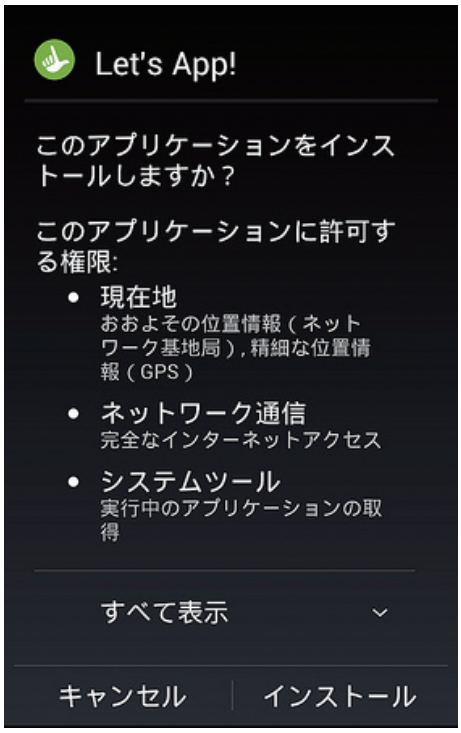

図 2.3: Android アプリケーションのパーミッション確認画面

しかし，見慣れてしまって確認が面倒になったり，表示されている内容が分かりにくいなどの理由で，ユーザによる事前チェックが形骸化しているという懸念がある．加えて，表示されている権限がどういった目的で使用されるのか記されておらず，総合的な作用や悪意の有無を，ユーザが判別できないといった問題も生じている.

また，Leviathan Security Group の発表 [22] によると，パーミッションを一切要求しない Android アプリケーションであっても，データを外部に送信できる可能性が指摘されている．他にも，独立行政法人情報処理推進機構（IPA）技術本部 セキュリティセンターのレポート [23] によると，他にインストールされているアプリケーションの脆弱性によって，ユーザが許可していないデータにアクセスされる恐れがあることが明らかにされている.

オペレーティングシステムのアップデート

Android には，現在まで様々な脆弱性が発見されたが，アップデートによって，対策がなされている．

しかし，Android は，Google の Nexus シリーズを除き，通信キャリアがオペレーティングシステムのアップデート時期をコントロールしているため，ユーザが望んでも，最新版へのアップデートができないといった問題がある．そのため，既知の脆弱性に対して対策がなされていない Android が，現在でも多く使用されている．Google が公開したデータ [24] によると，2010 年 12 月にリリースされたバージョン 2.3.3 以下の Android のシェアが未だに 60%以上あることが述べられている．

Version	Codename	API	Distribution
1.5	Cupcake	3	0.1%
1.6	Donut	4	0.3%
2.1	Eclair	7	2.7%
2.2	Froyo	8	10.3%
2.3 - 2.3.2	Gingerbread	9	0.2%
2.3.3 - 2.3.7		10	50.6%
3.1	Honeycomb	12	0.4%
3.2		13	1.2%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	27.5%
4.1	Jelly Bean	16	5.9%
4.2		17	0.8%

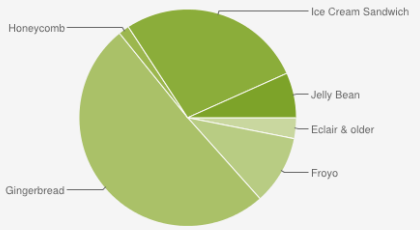

図 2.4: Data collected during a 14-day period ending on December 3, 2012

2.4.2 iOS におけるプライバシ侵害への対策

App Store の審査

App Store は，Google Play と異なり，アプリケーションの公開前に，Apple の厳格な審査を受ける必要がある．これにより，技術的・倫理的に問題のあるアプリケーションは排除され，ユーザが安心してアプリケーションを使用できる仕組みとなっている．

しかし，過去に悪意のある不正なアプリケーションが審査を通って公開された事例があり，また，プライバシを侵害するアプリケーションは見過ごされることが多く，App Store に公開されたままになっているのが現状である．

iOS6 のプライバシ設定

iOS はバージョン 6 から，アプリケーションからの個人情報へのアクセス許可を細かく設定できるようになった．位置情報，電話帳，カレンダー，リマインダー，写真，Bluetooth 共有の 6 点のデータアクセスに対して，アプリケーション毎に許可を与えるか設定するこ

とができる．これにより，ユーザはアプリケーションがユーザに無断で個人情報を送信しても，即座に察知できるようになった．

iOS6 のプライバシ設定画面例を図 2.5 に示す．

図 2.5: iOS6 のプライバシ設定画面例

UDID の使用の禁止

2.1.2 項で述べたように，UDID の使用にはプライバシ侵害の危険性が生じるため，Apple によって iOS での UDID の使用禁止が発表された．しかし，未だに多くのアプリケーションが UDID を使用し，また，禁止されていない MAC アドレスや IMEI を使用するアプリケーションも多く存在する．

2.4.3 ネットワークの検査ツール

ネットワークアナライザ

ネットワークアナライザは，同一 LAN 上を通過するトラフィックを監視したり記録するためのツールである．

代表的なソフトウェアとして，Wireshark が挙げられる．

Wireshark を使用することにより，スマートフォンアプリケーションの通信で送受信されたデータを閲覧し，プライバシを侵害していないかを検査することができる．

しかし，利用にはネットワークの知識が必要であり，一般ユーザには理解することが難しいといった欠点がある．Wireshark によるパケットキャプチャの様子を 2.6 に示す．

プロキシサーバ

プロキシサーバとは，クライアントとサーバ間の中継をするサーバである．

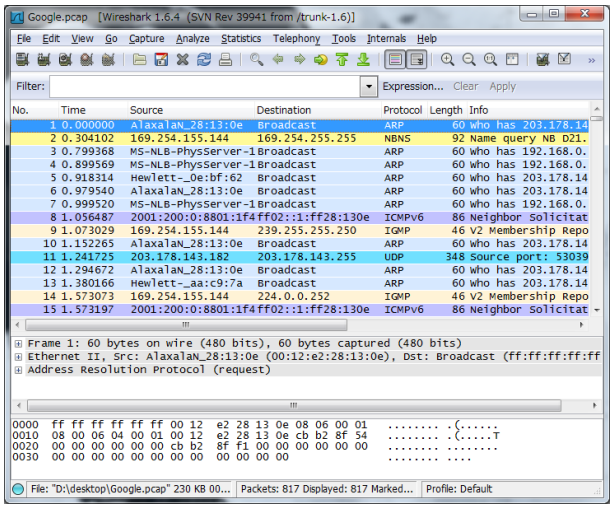

図 2.6: Wireshark によるパケットキャプチャ

情報元のサーバに対してはクライアントの情報を受け取り，クライアントに対してはサーバの働きをするため，ネットワークの検査ツールとしても利用できる．

また，中間者攻撃をすることによって，SSL/TLS で暗号化された通信も閲覧できるといった特徴がある．

ネットワークの検査のための代表的なプロキシサーバとして，Fiddler[25] が挙げられるが，Wireshark と同じく，ネットワークの知識がなければ理解が難しいといった欠点がある．Fiddler による中間者攻撃の様子を 2.7 に示す．

2.5 本論文の着眼点

2.2 節で述べたとおり，プライバシ面で不正な通信を行うスマートフォンアプリケーションには様々な種類があり，1 種類の検査だけでは対処できないという問題がある．また，2.4.3 項で述べた既存のネットワーク検査ツールには，検査結果を理解することが難しいといった欠点がある．本研究では，スマートフォンアプリケーションのプライバシ侵害を総合的に検査し，結果をネットワークの知識がないユーザであっても分かりやすく表示するプラットフォームを提案する．

2.6 まとめ

本章では，現状のスマートフォンにおけるプライバシの扱いについて示した．その後，プライバシ面で不正な通信を行うスマートフォンアプリケーションを 3 つに分類するとともに，それにより引き起こされる脅威について述べた．そして，ベンダによるプライバシ侵害への対策とネットワーク検査ツールを挙げ，その問題点について述べた．

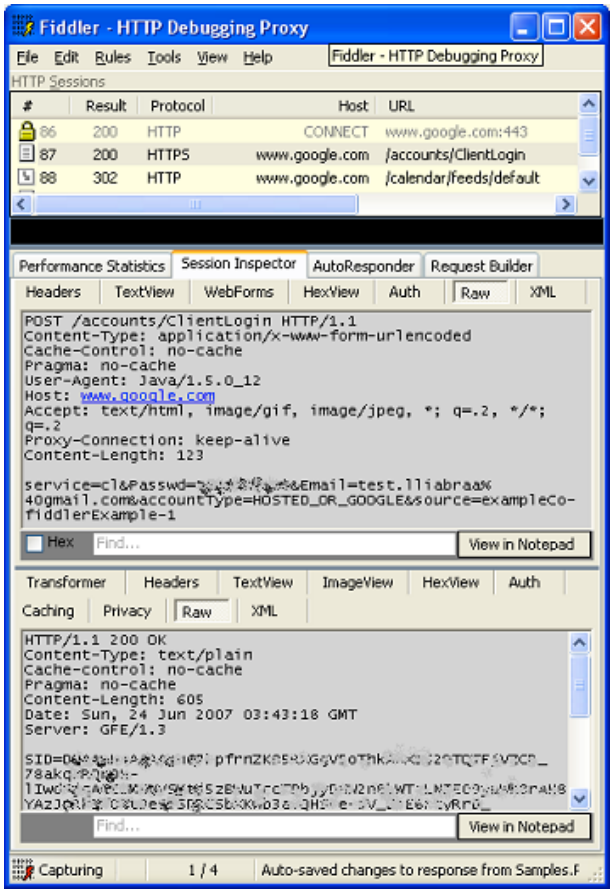

図 2.7: Fiddler による中間者攻撃

第3章　関連研究

本章では，スマートフォンアプリケーションを解析することによって，プライバシ面の不正を検知する既存研究について述べる．そして，それぞれの特徴を挙げ，問題点を指摘する．

3.1　静的解析によるアプリケーション検査

3.1.1　静的解析による Android アプリケーションの検査

”AndroidLeaks: Automatically Detecting Potential Privacy Leaks in Android Applications on a Large Scale”[26] では，APK(Android Package files) を静的解析し，プライバシ面で不正の動作をする Android アプリケーションを自動検知する手法を提案している．これによれば，24350 件の Android アプリケーションを 30 時間かけて検査して，7414 件のアプリケーションから，57299 件の潜在的なプライバシ侵害を検知することに成功している．

しかし，この手法では，個人情報を無断で送信するアプリケーションしか検査することができず，総合的なスマートフォンアプリケーションの検知システムとしては，不十分である．

AndroidLeaks の提案手法の概要を図 3.1 に示す.

3.1.2　静的解析による iOS アプリケーションの検査

”PiOS: Detecting Privacy Leaks in iOS Applications”[27] では，上記と同様に，静的解析によって，iOS アプリケーションのプライバシ面での不正を検査している．

この手法によって，1400 件の iOS アプリケーションを検査したところ，電話番号などの個人情報を無断送信するアプリケーションはほとんど検知できなかったが，多くのアプリケーションが個別識別子を送信していたことを明らかにしている．

しかし，この手法も AndroidLeaks と同様に，個人情報を無断送信するアプリケーションのみの検査であり，総合的なスマートフォンアプリケーションの検知システムとしては不十分である．

図 3.2 に PiOS のシステムの概要を示す.

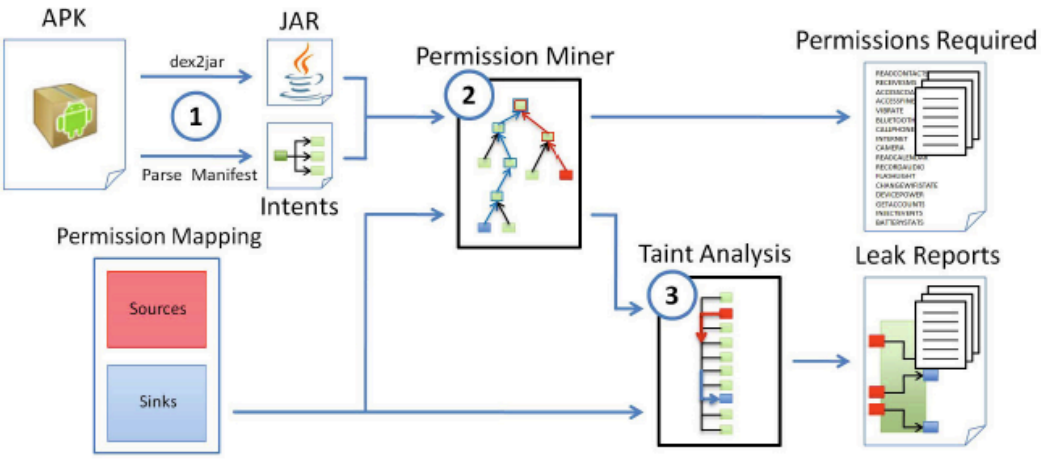

図 3.1: AndroidLeaks: Automatically Detecting Potential Privacy Leaks in Android Applications on a Large Scale, 2011, Gibler, Clint and Crussell, Jonathan and Erickson, Jeremy and Chen, Hao

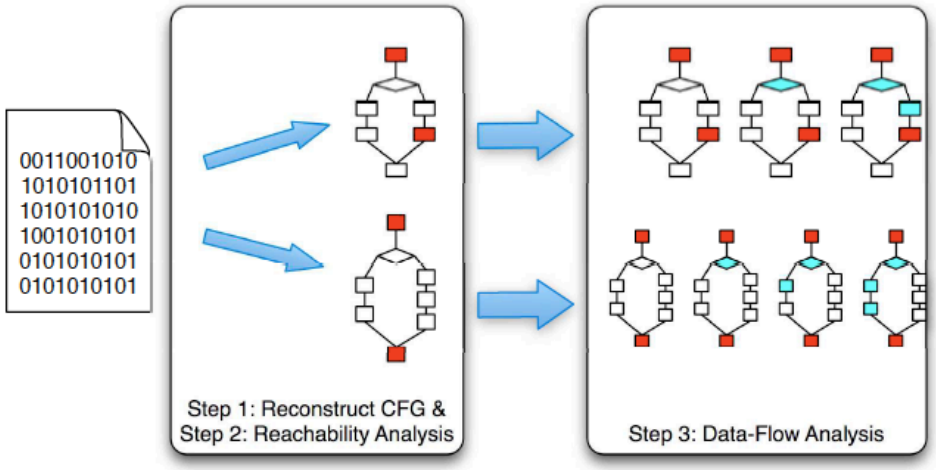

図 3.2: The PiOS system

3.2 アプリケーションの危険度の分類

Secroid[28] は，セキュリティベンダーのネットエージェントによって開発された，Android アプリケーションのリスクレベルの確認サイトである．このサービスでは，独自開発した自動判定エンジンを用いて，アプリケーションが使用している API やシグネチャから，DANGER，HIGH，MID，LOW，SAFE の 5 段階でアプリケーションのリスクレベルを評価し，アプリケーションが利用する情報の種類や，取得情報の送信先について掲載して

いる．

しかし，全ての Android アプリケーションが掲載されているわけではなく，最近公開されたアプリケーションは確認できないことが多い．また，Android の検査結果のみが掲載されており，iOS アプリケーションは確認ができないといった問題がある．

Secroid の表示画面を図 3.3 に示す．

図 3.3: Secroid の表示画面

第4章　通信の検証によるスマートフォンアプリケーションのプライバシ侵害検知システムの提案

本章ではまず，提案手法として，通信の検証によるスマートフォンアプリケーションのプライバシ侵害検知システムについて述べ，既存研究との違いや，本提案手法の利点を述べる．そして，本提案手法によって，どのようにスマートフォンアプリケーションのプライバシ侵害の検査が可能となるかを記述し，想定される本提案手法のユーザと利用例を述べる．また，本提案手法の実現に不可欠な機能要件を挙げ，その具体的な達成手法について述べる．

4.1　提案手法

スマートフォンアプリケーションのプライバシ侵害の検査のために，通信内容の検証による検知システムを提案する．本節では，提案手法と既存研究について述べ，その後，本提案手法の利点について述べる．

4.1.1　既存研究との違い

本研究と既存の研究の最大の違いは，スマートフォンアプリケーションの通信から，プライバシ侵害を検知する点である．既存研究のシステムでは，2.2.1 項や 2.2.3 項で述べた問題について検知できない．また，Android と iOS のどちらか一方のみの検査しかできないといった問題がある．

4.1.2　本提案手法の利点

本提案手法の利点として，2.2.1 項や 2.2.2 項，2.2.3 項のすべての問題に対して，検査できることが挙げられる．

また，SSL/TLS を用いて暗号化された通信であっても，本システムがプロキシとなり，中間者攻撃を行うことによって，通信内容を取得し，分析することが可能である．

既存のネットワーク解析ツールとしては，Wireshark や TCPDUMP が挙げられるが，本提案手法では，それらのツールと異なり，ネットワークの知識や手間を必要としないという利点がある．

加えて，通信ごとに危険度が色別に表示されるので，視覚的に理解が容易であり，Web ブラウザに結果が表示されるため，閲覧に OS の違いなどによる環境の制約がないことも，利点として挙げられる．

4.2 想定される対象者と利用例

本システムの主な使用対象者として，スマートフォンアプリケーションのプライバシの扱いについて懸念を抱いているが，ネットワークの知識が乏しいために，検査をすることができないユーザを挙げる．また，ネットワークについての知識は有しているが，スマートフォンアプリケーションの検査に，時間や手間がかけられないセキュリティ研究者も対象とする．

利用方法としては，本システムを自分の PC にインストールし，通信を取得した後，PC，もしくはスマートフォンの Web ブラウザを用いて，検査の結果を閲覧する方法が想定される．

4.3 要求事項

本項では，スマートフォンアプリケーションの通信検証システムを提案する上で，達成すべき事項を挙げる．要求事項として，平文で認証情報を送信する通信の検知，個人情報の無断送信の検知，中間者攻撃による SSL/TLS 通信の検査，ユーザインターフェースの明快さの 4 点があり，その内容について詳しく述べる．

4.3.1 平文で認証情報を送信する通信の検知

平文で認証情報を送信するスマートフォンアプリケーションの存在と，プライバシ上の問題について，2.2.1 項で述べた．この問題に対処するため，本提案手法では，平文での認証情報の送信を検知し，ユーザに警告を出す必要がある．

具体的には，HTTP 通信から，GET・POST のパラメータや Cookie などの項目を読み取り，認証情報を含んでいるかを分析する．

また，それらの文字列は，BASE64 でエンコードされている可能性があるため，デコードする処理が必要となる．

4.3.2 個人情報の無断送信の検知

2.2.2 項で述べた問題に対処するため，本提案手法において，個人情報がユーザに無断で送信されていないかを検知する必要がある．

本研究では，以下の 6 点の個人情報を検知対象とする．

- 電話番号
- メールアドレス
- 個体識別情報 (Android ID，IMEI，MAC アドレスなど)
- 位置情報
- パスワード
- 年齢・性別

これらの個人情報が通信に含まれているか，文字列のパターンマッチを行うことによって分析しなければならない．

また，送信先ホストによる判別も行う必要がある．具体的には，以下の 2 点で判別する．

- 広告会社のホストに送信しているもの
- 前後の通信と関係のないホストに送信しているもの

4.3.3 中間者攻撃による SSL/TLS 通信の検査

上記で述べたように，個人情報の無断送信を検知する必要があるが，SSL/TLS で暗号化された通信は，通常の方法では中身を取得することができない．そのため，中間者攻撃を行うことにより，SSL/TLS 通信の中身を検査する必要がある．

また，この方法により，2.2.3 項で述べた，中間者攻撃への脆弱性があるアプリケーションを調べることが可能である．これは，偽の証明書を使用していた場合に，スマートフォン側で，アプリケーションが適切にエラーとするか否かで判別できる．

4.3.4 ユーザインターフェースの明快さ

通信内容の分析といった，手間をかけずに直感的に結果を理解できるようにする必要がある．そのため，プライバシ侵害への危険度を分類し，パケットごとに色別で表示する．

また，ネットワークの知識がないユーザであっても，容易に理解できるように，どういった種類の情報を，どこのホストに対して送信しているのかといった詳細な情報についても分かりやすく表示する必要がある．

加えて，ユーザが即座に結果を取得できるように，検査結果は画面にリアルタイムに表示する必要がある．

他にも，検査結果を Web ブラウザで閲覧できるようにすることで，使用するデバイスなどの環境に関係なく，結果を閲覧できるようにしなければならない．

4.4 まとめ

本章では，提案手法について具体的な内容を述べ，既存の研究との違いと，本提案手法を用いることの利点について記述した．その後，本提案手法の想定されるユーザと利用例を述べ，本提案手法が達成すべき 4 つの要求事項を挙げた．

第5章　実装

本章では，4 章で述べた手法を用いて，本研究で開発したシステムの具体的な設計と実装について述べる．まず，4.3 節で述べた要求事項を元に，本システムの設計概要について記述する．その後，実装環境について述べ，実際に実装した実装物の概要について記す.

5.1　設計

本研究は，スマートフォンアプリケーションの通信を自動で分析し，判定するシステムを実装する．図 5.1 では，本実装の設計概要を示している．また，本システム実装環境を表 5.1 に示す.

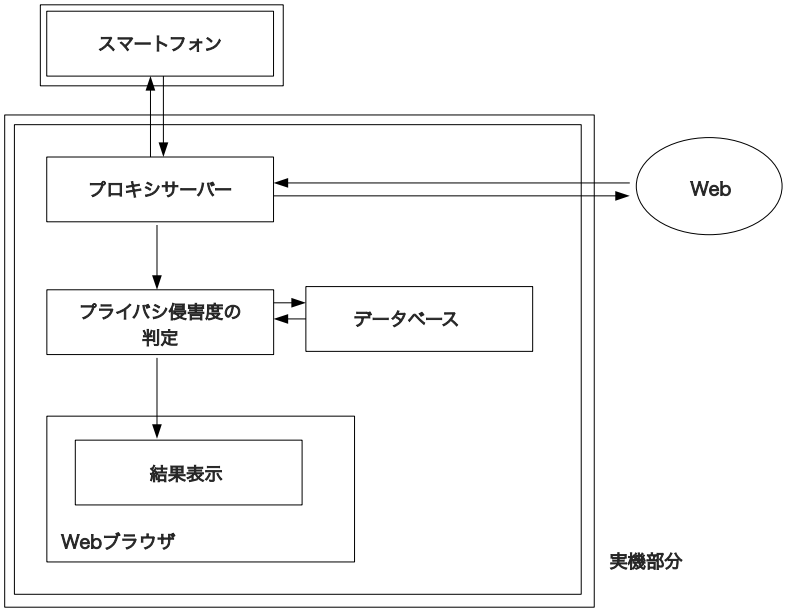

図 5.1: 設計概要

表 5.1: 実装環境

要素名	属性	利用環境
OS	実機	Mac OSX Moutain Lion 10.8[29]
言語		Ruby version 1.9.3p194[30]
データベース		MongoDB version 2.2.0[31]
ライブラリ	プロキシサーバ	Webrick version 1.3.1[32]
	Web フレームワーク	Sinatra version 1.3.3[33]
	SSL/TLS 証明書生成	Quickcert version 1.0.2[34]
	プロセス管理	Foreman version 0.60.2[35]
	Ajax 処理など	jQuery version 1.8.3[36]

5.2 システム構成

本節では，実装したシステムの構成について詳しく述べる本システムは，プロキシサーバ部分，解析部分，結果出力部分の 3 つの部分から構成される．

プロキシサーバ部分は，スマートフォンアプリケーションとインターネットの中継を行うことよって，パケット単位でデータを読み込み，そこから必要な情報のみを抽出し，解析部分に渡す．詳細は，5.2.1 項で述べる．

解析部分では，プロキシサーバ部分から受け取った情報を解析し，結果出力部分にデータを渡す．詳細は，5.2.2 項で述べる．

結果出力部分は，解析部分から受け取った結果を Web ブラウザ上に分類して，表示する部分である．詳細は，5.2.3 項で述べる．

システム構成の概要を図 5.2 に示す.

5.2.1 プロキシサーバ部分

本項では，スマートフォンアプリケーションとインターネットの中継を行うことによって，パケットを取得し，必要な情報のみを抽出するプロキシサーバ部分について述べる．また，SSL/TLS で暗号化されている通信においては，中間者攻撃を行うことによって，パケットを取得する．

対象プロトコル

HTTP，HTTPS の通信をキャプチャ対象とする．

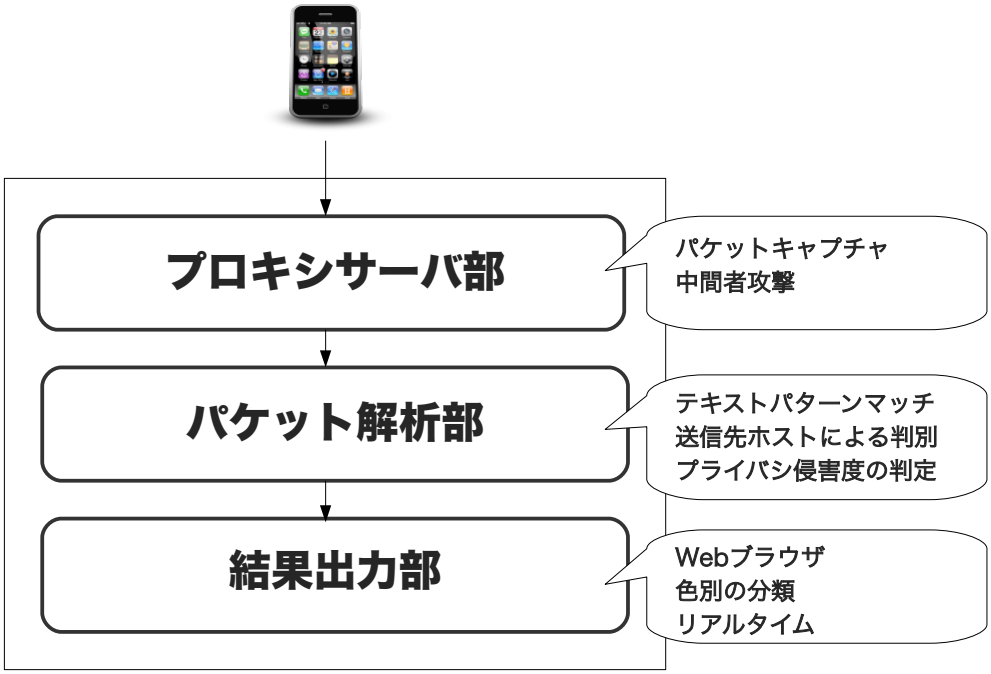

図 5.2: システム概要

取得情報

HTTP のヘッダとボディから情報を抽出する．取得する情報は，送信先ホスト，HTTP メソッド，Cookie，GET・POST のクエリ，SSL/TLS による暗号化の有無，リクエストファイルの種類などが挙げられ，解析部分に渡されて使用される．

中間者攻撃による SSL/TLS 通信の閲覧

本システムでは，スマートフォンアプリケーションと送信先サーバに，正規の SSL/TLS 通信と見せかけることによって，中間者攻撃を行う．そのため，あらかじめユーザによって，準備として，検査対象のスマートフォンに，本システムが発行するルート証明書をあらかじめインストールしてもらう必要がある．

プロキシサーバ部分では，送信先ホストごとに，偽のサーバ証明書を生成する．これによって，暗号化通信を復号し，通信内容を取得する．

概要を図 5.3 に示す．

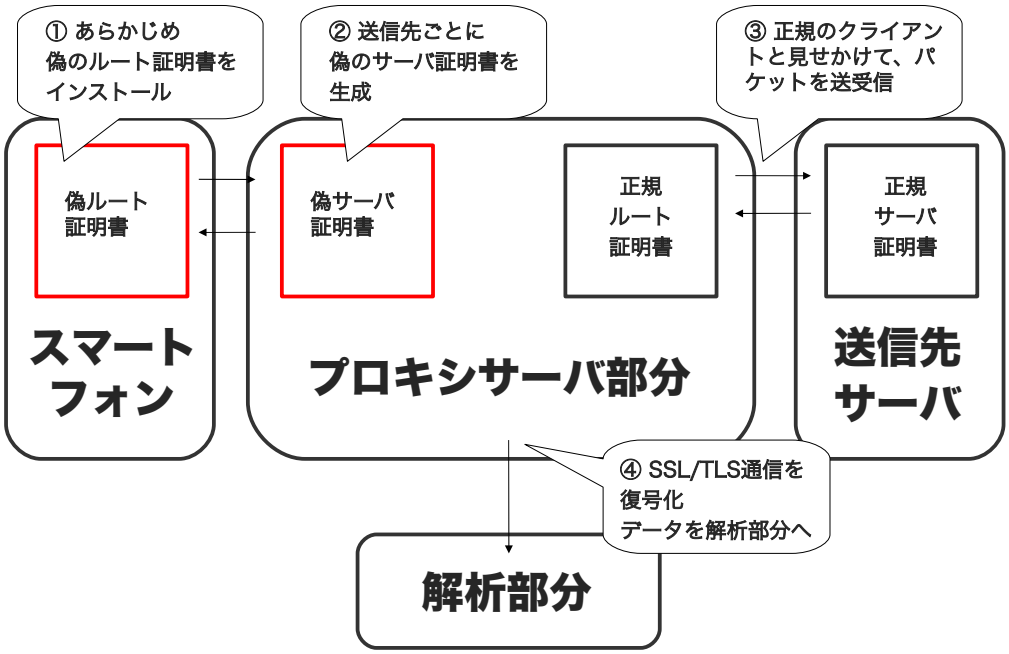

図 5.3: プロキシサーバー部分概要

5.2.2 解析部分

解析部分は，プロキシサーバ部分で取得したデータを解析し，スマートフォンアプリケーションが，プライバシを侵害する通信を行なっているかを検証する部分である．

文字列パターンマッチ

個人情報や認証情報を送信していないかを，文字列のパターンマッチによって検査する．具体的には，GET・POST のクエリなどの項目に passwd や android_id などといった文字列を含んでいるか分析し，電話番号やメールアドレスは正規表現によって判定する．また，ユーザがあらかじめ登録した単語や正規表現についても，同様に分析し，判定を行う．

送信先ホストによる判別

送信先ホストを，あらかじめデータベースに登録したホスト 50 件と照合し，一致した場合に，利用する情報からプライバシ侵害度を判定する．

また，ホストのテーブル構造を表 5.2 に示す．

表 5.2: ホストのテーブル構造

カラム	データタイプ	デフォルト値
ホスト名	String	NULL
利用する情報	Hash	NULL
所在国	String	NULL

通信内容の保存

通信内容は全てデータベースに記録され，次回以降のプライバシ侵害の判定材料や，結果表示画面の検索機能に利用する．表 5.3 にテーブル構造を示す．

表 5.3: HTTP 通信のテーブル構造

カラム	データタイプ	デフォルト値
ID	Integer	NULL
Timestamp	DateTime	NULL
プロトコル	String	”HTTP”
メソッド	String	GET
ホスト	String	NULL
HTTP Header	Hash	NULL
HTTP Body	String	NULL
プライバシ侵害度	Integer	1

プライバシ侵害度の分類

プライバシ侵害度の判定基準と，検査結果部分での表示色について以下に示す．

- Level1（灰色）
  - 普通の HTTP 通信
- Level2（緑色）
  - SSL/TLS で暗号化された認証
- Level3（黄色）
  - 個別識別番号の送信

- Level4（赤色）
  - 平文によるパスワードの送信
  - メールアドレスの送信
- Level5（黒色）
  - SSL/TLS 処理に不備があるもの
  - 禁じられた動作を行うホストへの送信
  - 電話番号の送信
  - 位置情報の送信
  - ユーザ登録単語の送信

5.2.3 結果出力部分

結果表示部分は，検証結果を Web ブラウザ上に表示する部分である．結果表示画面例を図 5.4 に示す.

図 5.4: 結果表示画面例

リアルタイム表示

解析部分から結果表示部分への送信時のプロトコルに Websocket を用いて，また，Web ブラウザで Ajax によって画面書き換えを行うことによって，Web ブラウザ上へリアルタイムに結果を表示する．

検査単語・正規表現の登録

Web ブラウザより，ユーザが検査したい単語や正規表現を登録する．登録単語は，MD5 や SHA-1 でハッシュ値を計算し，ハッシュ化した文字列もパターンマッチの対象とする．図 5.5 に登録画面を示す．

図 5.5: 単語・正規表現登録画面

位置情報の取得

位置情報の文字列パターンマッチを行うために，Web ブラウザにおいて，あらかじめ位置情報を取得する．

5.2.4 設定と実行

本システムでは，ポート番号の指定や，検査除外ホストなどの項目を設定ファイルである config.yml において指定する．設定項目一覧を表 5.4 に示す．
また，本システムを実行するためのコマンドを図 5.6 に示す．

図 5.6: 実行コマンド

```
rake
```

表 5.4: 設定項目

項目	デフォルト値	説明
proxy_port	9999	プロキシサーバのポート番号
result_port	5000	結果表示部分のポート番号
websocket_port	3456	結果送信用 websocket のポート番号
inspect_ssl	true	SSL/TLS 通信の検査をするか否か
exclude_host	[]	検査から除外するホスト

5.3 まとめ

本章では，4 章で述べた提案手法に基づいて，実装したシステムの設計と想定される利用環境について述べた．その後，システム構成として，プロキシサーバ部分，解析部分，結果表示部分の 3 つの部分について，詳細を明らかにするとともに，本システムの設定と実行方法を記述した．

第6章　提案システムを用いたアプリケーションの検査と考察

本章では，5章で実装したシステムを用いて，実際に市場のスマートフォンアプリケーションに対して検査を行う．その後，検査の結果から，本研究に対する考察を行う．

6.1　提案システムを用いたアプリケーションの検査

本節では，実際に提案システムを使用して，市場で公開されているスマートフォンアプリケーション100件を対象に，プライバシを侵害するような動作を行なっていないか検査を行う．

6.1.1　検査の概要

検査の概要を以下に示す．

- 期間
  2013年1月20日-2013年1月22日
- 検査対象
  Google Playに公開されている人気無料Androidアプリケーション50件
  App Storeに公開されている人気無料iOSアプリケーション50件
- 使用機器とオペレーティングシステム
  Galaxy Nexus, Android 4.1.1
  iPhone 3GS, iOS 5.1

6.1.2　検査結果

本節では，上記で示した方法を用いて行った検査について，Androidアプリケーションと iOSアプリケーションの2つの結果に分けて述べる．

表 6.1: Android アプリケーションの検査結果

項目	送信情報	件数	比率
平文で認証情報を送信するアプリケーション		3 件	6%
個人情報を送信するアプリケーション	電話番号	2 件	4%
	メールアドレス	1 件	2%
	位置情報	3 件	6%
	個別識別子	24 件	48%
	年齢・性別	0 件	0%
SSL/TLS の処理に不備があるアプリケーション		3 件	6%

Android アプリケーションの検査結果

人気 Android アプリケーション 50 件に対する検査結果を表 6.1 に示す.
また，以下にそれぞれの項目に対する詳しい検査結果を述べる.

- 平文で認証情報を送信するアプリケーション
  ゲームアプリケーションにおいて，関連サービスへのログイン時に，パスワードを平文で送信していた．また，平文で送信していても，パスワードをハッシュ化しているものがあった.

- 個人情報を送信するアプリケーション
  個人情報を送信していたアプリケーション 26 件のうち，8 件が利用規約に個人情報を利用する旨を表示していた．しかし，18 件が利用規約に使用する旨を書いていないか，利用規約自体がなかった．更に，22 件が個人情報を広告業者やアクセス解析業者に送信し，6 件がアプリケーションデベロッパや自社のサーバに送信していた.

  また，サービス規約が表示される前に，メールアドレスを無断で送信していたアプリケーションが 1 件あったが，電話番号を送信していたアプリケーションは，事前にユーザの同意を得ていた.

- SSL/TLS の処理に不備があるアプリケーション
  検知した 3 件のうち，1 件は Web ブラウザであり，証明書が一致しなくても，読み込む全ての Web サイトで，警告を全く表示しなかった．また別の 1 件は，エラーダイアログが表示されたが，中間者攻撃には成功しており，パスワードが閲覧可能であった.

iOS アプリケーションの検査結果

人気 iOS アプリケーション 50 件に対する検査結果を表 6.1 に示す.
また，以下にそれぞれの項目に対する詳しい検査結果を述べる.

表 6.2: iOS アプリケーションの検査結果

項目	情報の種類	件数	比率
平文で認証情報を送信するアプリケーション		2 件	4%
個人情報を送信するアプリケーション	電話番号	3 件	6%
	メールアドレス	0 件	0%
	位置情報	4 件	8%
	個別識別子	17 件	34%
SSL/TLS の処理に不備があるアプリケーション		0 件	0%

- 平文で認証情報を送信するアプリケーション
  カメラアプリケーションにおいて，撮影した写真を SNS で共有する際に，SNS のログインパスワードが平文で送信されていた．
- 個人情報を送信するアプリケーション
  個別識別子を送信していた 17 件のうち，13 件が UDID や MAC アドレスを送信していた．そのうち，6 件は MD5 や SHA-1 によって，ハッシュ化されていた．
- SSL/TLS の処理に不備があるアプリケーション
  今回の検査では，SSL/TLS の処理に不備がある iOS アプリケーションは検知できなかった．

6.2 考察

本節では，上記の検査の結果を受け，不正な通信を行う通信の種類や，送信する情報の種類ごとに考察を行う．

6.2.1 平文で認証情報を送信するアプリケーションの考察

平文で認証情報を送信するアプリケーションの検知については，通信の検証が不可欠であり，本システムが有効に働いた．しかし，平文であっても，ハッシュ化されているものは，本システムでは検知が不可能であり，ユーザに提示して判定してもらう必要があった．

6.2.2 個人情報を無断で送信するアプリケーションの考察

電話番号

正規表現による電話番号の検知では，ランダムな数値との誤検知が 1 件あったが，また，利用規約に同意した上での，送信

メールアドレス

今回，検知した Android アプリケーションは，利用規約の前にメールアドレスを無断で送信していた．本システムでは，通信を解析しているため，情報を送信するタイミングを知ることができ，今回の無断送信の検知につながった．

位置情報

Web ブラウザ上であらかじめ取得した位置情報を使って，位置情報の検知が可能であった．

個別識別子

UDID や IMEI などの個別識別子は，MD5 や SHA-1 などによってハッシュ化されたものも含まれており，目視による検査では非常に手間がかかるが，本システムでは容易に検知することができた．

6.2.3 SSL/TLS の処理に不備があるアプリケーションの考察

SSL/TLS で暗号化される通信は，機密情報を含んでいることが多く，脆弱性検査の必要性が高い．特に，今回のように，Web ブラウザに中間者攻撃の脆弱性が存在した場合，全ての Web サイトの暗号化通信に対して盗聴や改竄の恐れがある．本システムによって検知できたことは，先行研究に対する優位点であると言える．

6.3 まとめ

本章では，5 章で実装したシステムを，実際に市場のスマートフォン 100 件に対して使用し，検査を行った．その後，得られた結果から，提案手法の有効性や，本研究に対する考察を行った．

第7章　結論

本章では，本論文の全体についてまとめ，1.3 節で述べた目的の中で，本研究によって達成された事柄について述べる．そして，本研究の発展を実現するために，今後の展望を述べる．

7.1　まとめ

本研究の目的は，スマートフォンアプリケーションがプライバシを侵害するような動作を行なっているか，容易に検査が可能な環境の実現である．これにより，ユーザは，利用するアプリケーションのプライバシ面での危険度を把握でき，より安心してスマートフォンを利用できる．

そこで，本論文では上記のような目的を達成するために，スマートフォンアプリケーションの通信内容を検証することによって，プライバシ侵害度を判定し，結果をユーザに分かりやすく表示するプラットフォームを提案した．

そして，この提案手法に基づいて，システムを設計・実装し，その実装したシステムを用いて，実際に市場に公開されている人気スマートフォンアプリケーション計 100 件に対して，検査を行った．

その結果，36 件のアプリケーションが，プライバシ侵害を行なっていたことが判明し，本提案手法で，十分検知できることが実証された．

7.2　今後の展望

本節では，本論文全体のまとめを受けて，本論文の発展のため，今後の展望について述べる．

7.2.1　判定の精度

現時点の判定では，誤検知やプライバシ侵害の見逃しが発生している．これは，実際に，プライバシ侵害があった通信に多い文字列パターンを学習することによって，検知率を向上させることが可能であると推測する．

また，個人情報が無断に送信されたものなのか，同意の上で送信されたものなのか，本システムでは判定できず，ユーザの判断に任せる必要がある．

7.2.2 PC アプリケーションの検査

本研究では，スマートフォンアプリケーションの通信内容の検証に特化したシステムを構築した．

しかし，T ポイントツールバー [37] が WEB 閲覧履歴を平文で送信していたように，PC においてもスマートフォンと同様に，プライバシを侵害するアプリケーションに対しての懸念が広がっている．

そこで，本研究における提案手法の判定方法を PC アプリケーション用に修正し，PC 用の検査プラットフォームを構築することによって，これらの問題を解決することが期待される．

謝辞

本論文の作成にあたり，ご指導頂いた慶應義塾大学環境情報学部学部長 村井 純博士，同学部教授 徳田 英幸博士，同学部教授 中村 修博士，同学部准教授 楠本 博之博士，同学部准教授 高汐 一紀博士，同学部准教授 三次 仁博士，同学部准教授 植原 啓介博士，同学部専任講師 中澤 仁博士，同学部准教授 Rodney D.Van Meter III 博士，同学部教授 武田 圭史博士，同大学政策・メディア研究科特任講師 斉藤 賢爾博士，同研究科特別研究講師 佐藤 雅明博士に感謝致します．特に武田圭史博士は，研究で行き詰まる私に対して非常に根気強く指導して下さり，常に新しいアイディアと研究手法で私を導いていただくことで，何度も私に新しい視点や手本を見せていただきました．本当にありがとうございました．

そして，本研究を進めていく上で，様々な励ましと助言，お手伝いをいただきました，村井研究室卒業生である，上原 雄貴氏，重松 邦彦氏，梅田 昇翔氏，福岡 英哲氏，相見 眞男氏，Doan Viet Tung 氏，Vu Xuan Duong 氏，Pham Van Hung 氏，吉原 洋樹氏に感謝致します．

慶應義塾大学政策・メディア研究科修士課程，碓井 利宣氏，関根 冬輝氏，山本 知典氏に感謝致します．研究手法や，論文の書き方，研究発表の資料のレビューなど，研究に関わるあらゆる事項を熱心に指導していただきました．彼らの協力なしには，本研究を進めることはできませんでした．心から感謝いたします．

研究室で苦楽を共にした有馬 怜文氏，石野 佑樹氏，大野 三津雄氏，大矢 崇央氏，恩田 優女史，Nguyen Anh Tien 氏，鴻野 弘明氏，高岡 賢二氏，中島 明日香女史，星出 直柔氏，三ツ木 あかね女史，由井 卓哉氏，吉原 大道氏，生方 悠介氏，小澤 麗女史，Tran Ngoc Anh 氏，露木 航平氏，中安 恒樹氏，深谷 哲史氏，岡田 英晃氏，川本 卓弥氏，芹澤 親氏，廣田 一貴氏，八木橋 優氏，山田 夏才氏に感謝致します．彼らと一緒に研究をすることでお互いを刺激しあい，より質の高い議論や研究をすることができました．また，体外活動やレクリエーション，まるたか，ISC 焼肉の会，就職活動など，研究以外の様々な面でもお世話になりました．この場を借りてお礼を述べさせていただきます

私の大学 4 年間の心の拠り所であったアカペラサークル K.O.E. とサークル員全員に心から感謝いたします．特に，同期である 09 メンバーのおかげで，余裕を持った学生生活及び研究活動を行うことができたと確信しております．

私のアルバイト先であるファミリーマート藤沢慶応大学前店，及び店長を始めとしたスタッフ全員に感謝いたします．午前 6 時からの早朝アルバイトのおかげで，規則正しい生活を送ることができ，ほとんど体調を崩すことなく研究に力を注ぐことができました．また，各発表当日にアルバイトをすることによって，爽やかな気分で研究発表をすることができました．

最後に，大学入学からの 4 年間だけでなく 23 年間をあらゆる面で支えていただいた父，小松 松太郎，母，小松 喜美と私の家族に心から感謝致します.

参考文献

[1] シード・プランニング. 世界のスマートフォン普及予測. http://www.seedplanning.co.jp/press/2012/2012072601.html, 7 2012.

[2] Google Inc. Android. http://www.android.com/, 10 2008.

[3] Apple Inc. ios. http://www.apple.com/jp/ios/, 6 2007.

[4] Appsfire SAS. Appsfire. http://appsfire.com/, 10 2012.

[5] Apple Inc. App store. http://www.apple.com/jp/iphone/from-the-app-store/, 10 2012.

[6] BusinessWeek. Google says 700,000 applications available for android. http://www.businessweek.com/news/2012-10-29/google-says-700-000-applications-available-for-android-devices, 10 2012.

[7] Google Inc. Google play. https://play.google.com, 10 2012.

[8] Steven McCanne Van Jacobson, Craig Leres. tcpdump. http://www.tcpdump.org/, 1987.

[9] Gerald Combs. Wireshark. http://www.wireshark.org/, 6 2006.

[10] University of Ulm. The insecurity of google's client login protocol. http://www.uni-ulm.de/en/in/mi/staff/koenings/catching-authtokens.html, 5 2011.

[11] Instagram. Instagram. http://instagram.com/, 12 2012.

[12] Inc Foursquare Labs. foursquare. https://ja.foursquare.com/, 12 2012.

[13] 独立行政法人情報処理推進機構 技術本部 セキュリティセンター. 情報セキュリティの脅威に対する意識調査 報告書, 12 2011.

[14] 竹森 敬祐. スマートフォンからの利用者情報の送信 ～情報収集の実態調査～, 2012.

[15] SCOTT THURM and YUKARI IWATANI KANE. Your apps are watching you. http://online.wsj.com/article/SB10001424052748704694004576020083703574602.html, 10 2012.

[16] Sascha Fahl, Marian Harbach, Thomas Muders, Lars Baumgärtner, Bernd Freisleben, and Matthew Smith. Why eve and mallory love android: an analysis of android ssl (in)security. In *Proceedings of the 2012 ACM conference on Computer and communications security*, CCS '12, pages 50–61, New York, NY, USA, 2012. ACM.

[17] Dave Morin. Path. `https://path.com/`, 11 2012.

[18] Unknown. ios 版の path は ssl の証明書エラーを無視している. `http://subtech.g.hatena.ne.jp/mala/20120214/1329196803`, 2 2012.

[19] Official Google Mobile Blog. Android and security. `http://googlemobile.blogspot.jp/2012/02/android-and-security.html`, 2 2012.

[20] Charlie Miller Jon Oberheide. Dissecting the android bouncer. `http://jon.oberheide.org/blog/2012/06/21/dissecting-the-android-bouncer/`, 6 2012.

[21] Google Inc. Android security and permissions. `http://developer.android.com/guide/topics/security/permissions.html`, 12 2012.

[22] Leviathan Security Group. Zero-permission android applications. `http://leviathansecurity.com/blog/archives/17-Zero-Permission-Android-Applications.html`, 4 2012.

[23] 独立行政法人情報処理推進機構 技術本部 セキュリティセンター. 「android アプリの脆弱性」に関するレポート, 6 2012.

[24] Google Inc. Platform versions - android developer dashboard. `http://developer.android.com/about/dashboards/index.html#Platform`, 12 2012.

[25] Eric Lawrence. Fiddler. `http://www.fiddler2.com/fiddler2/`, 10 2003.

[26] Clint Gibler, Jonathan Crussell, Jeremy Erickson, and Hao Chen. Androidleaks: Automatically detecting potential privacy leaks in android applications on a large scale. In Stefan Katzenbeisser, Edgar Weippl, L.Jean Camp, Melanie Volkamer, Mike Reiter, and Xinwen Zhang, editors, *Trust and Trustworthy Computing*, volume 7344 of *Lecture Notes in Computer Science*, pages 291–307. Springer Berlin Heidelberg, 2012.

[27] Manuel Egele, Christopher Kruegel, Engin Kirda, and Giovanni Vigna. PiOS: Detecting privacy leaks in iOS applications. In *Proceedings of the 18th Annual Network & Distributed System Security Symposium (NDSS)*, February 2011.

[28] ネットエージェント株式会社. secroid. `http://secroid.jp/`, 10 2012.

[29] Apple Inc. Mac osx moutain lion 10.8. `http://www.apple.com/jp/osx/`, 2012.

[30] まつもとゆきひろ. Ruby version 1.9.3p194. http://www.ruby-lang.org/, 1995.

[31] 10gen. Mongodb version 2.2.0. http://www.mongodb.org/, 2009.

[32] 後藤裕蔵 高橋征義. Webrick version 1.3.1. http://www.webrick.org, 2003.

[33] Blake Mizerany. Sinatra version 1.3.3. http://www.sinatrarb.com/, 9 2007.

[34] Hiroshi Nakamura. Quickcert version 1.0.2. http://segment7.net/projects/ruby/QuickCert/, 2004.

[35] David Dollar. Foreman version 0.60.2. https://github.com/ddollar/foreman, 5 2010.

[36] John Resig. jquery version 1.8.3. http://jquery.com/, 8 2006.

[37] Culture Convenience Club. T ポイントツールバー. https://tsite.jp/toolbar/index.pl, 9 2012.