Androidアプリケーションにおける情報収集モジュール通信判別手法

葛野 弘樹[1,a)]　潤潟 謙一[1,b)]

概要：Androidを搭載した端末において，第三者の開発したAndroidアプリケーションが幅広く利用されている．アプリケーションのうち，無料アプリケーションの多くは，広告配信や利用統計の利用を目的とした情報収集モジュールを組み込んでいる．この情報収集モジュールは，端末の利用者に関する情報を収集して，利用者の意図しない情報漏えいを起こす可能性があると指摘されている．我々は，1,188個のアプリケーションから収集したHTTP通信を調査し，797個のアプリケーションにおいて，46個の既知の情報収集モジュールの通信が含まれていることを確認した．さらに，情報収集モジュールの通信を解析し，広告画像の取得に至るまでに一連の規則があることが判明した．我々は，アプリケーションの通信に，このような情報収集モジュールによる通信が含まれているかを検知するために，問題があるとされる情報収集モジュールのHTTP通信をグラフ化し，グラフ間の距離に基づく検知手法を提案する．提案手法の評価として，情報収集モジュールのHTTP通信のグラフ4,696個の一部を用いた場合において，情報収集モジュールのHTTP通信に対して72%の検知精度を示すことを確認した．

キーワード：Android, スマートフォン, プライバシー保護，グラフ解析

Detection Method of Advertisement Module Traffic in Android Application

KUZUNO HIROKI[1,a)]　MAGATA KENICHI[1,b)]

1. はじめに

携帯端末やタブレットなどのOSとしてGoogle Inc.のAndroidを搭載した機器が普及している．Androidでは，第三者の開発したAndroidアプリケーション（以下，アプリケーション）をGoogle Playなどのマーケットを通じて端末の利用者（以下，利用者）に提供できる．2013年2月時点で，800,000を超えるアプリケーションがマーケットにて公開されている[1].

マーケットのアプリケーションは，有料か無料に分類される．我々はそのうち無料アプリケーションに着目する．無料アプリケーションの多くは，利用者へのターゲット広告の配信や利用統計のための情報収集モジュール（Advertisement module）を含んでおり，これにより，アプリケーションの開発者が情報収集モジュールの提供者より広告収入を受け取るビジネスモデルが成立している．情報収集モジュールは，端末における利用者に関する情報にアクセスし，外部に送信することから，利用者の意図しない情報漏えいが起こる可能性が指摘されており[2], [3], [4]，Graceらの調査では，端末上の情報を利用する情報収集モジュールは100個以上確認されている[5].

Androidは，端末上の情報やリソースに対するアクセス制御にパーミッションフレームワークを用いており，アプリケーションの実行中におけるアクセス対象毎に必要なパーミッションをあらかじめ保持しておかねばならない．情報収集モジュールはアプリケーションにライブラリとして内包されるため，アプリケーションが提供する機能が要求するパーミッションと，情報収集モジュールが要求するパーミッションを実行前に区別することは困難となっている．そのため，アプリケーションから情報収集モジュールを分離し，より細かなアクセス制御を行う手法[6], [7]が提案されている．しかし，これらの手法は，利用者にAndroid

1 セコム株式会社IS研究所
Intelligent Systems Laboratory, SECOM Co., Ltd., Mitaka, Tokyo, 181–8528, Japan
a) h-kuzuno@secom.co.jp
b) k-magata@secom.co.jp

の修正およびパーミッションの正確な理解を必要とする．

我々は，アプリケーションの通信からシグネチャを生成し，Android を修正せずに端末上の情報送信を未然に防ぐ手法を提案している [8]．さらなる，アプリケーションによる利用者に関する情報の漏えい防止において，情報収集モジュールの振る舞いを正確に把握し，アプリケーションにどのような情報収集モジュールが含まれているかを特定することは重要な課題といえる．

本稿では，そのような課題を解決するため，アプリケーションの HTTP 通信に含まれる情報収集モジュールの HTTP 通信を検知する手法を提案する．提案手法では，情報収集モジュールによる広告画像の取得に HTML や JavaScript を利用した一連の規則があることに着目することで HTTP 通信をグラフ化し，任意の 2 つの HTTP 通信のグラフの距離を計算することで，情報収集モジュールの HTTP 通信のグラフを基に，アプリケーションに含まれる情報収集モジュールの HTTP 通信を検知する．

我々は，1,188 個のアプリケーションより収集した HTTP 通信をデータセットとして利用し，提案手法によりグラフ化した情報収集モジュールの HTTP 通信を利用し，アプリケーションの HTTP 通信に対する情報収集モジュールの HTTP 通信の検知精度の評価を行った．

本稿では，以下，2 章にて Android パーミッションフレームワークと情報収集モジュールの概要ついて，3 章にて情報収集モジュールの課題および実際の通信内容について述べる．4 章にて提案手法である HTTP 通信のグラフ化，グラフの距離を用いた情報収集モジュールの HTTP 通信の検知手法について述べ，5 章にて，提案手法の評価と評価結果の考察を行う．6 章で関連研究と提案手法の比較を行い，今後の課題について述べ，最後に，7 章で全体のまとめについて述べる．

2. 背景知識

2.1 パーミッションフレームワーク

Android は，アプリケーションの権限管理にパーミッションフレームワークを提供している．パーミッションは，Android における情報やリソースと紐付けられており，アプリケーションがそれらにアクセスする際には該当するパーミッションが必要となる．Android API Level 17 におけるパーミッションは 130 個 [9] 定義されている．パーミッションの例として，**INTERNET** パーミッションは，アプリケーションにソケットを利用したネットワーク通信を許可する．**READ_PHONE_STATE** パーミッションは，アプリケーションに端末の IMEI (International Mobile Equipment Identity) 等の端末識別番号や電話番号へのアクセスを許可する．また，パーミッションは，開発者が独自に定義することもでき，アプリケーション間で連携して動作する際のアクセス制御にも利用できる．

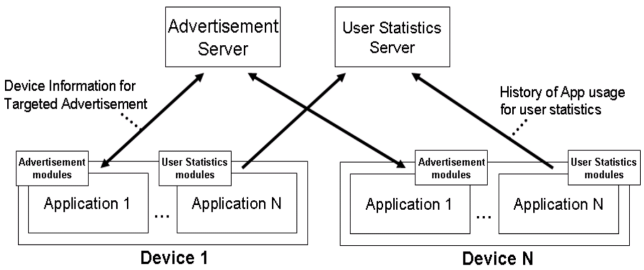

図 1 情報収集モジュールによる通信の概要．アプリケーションに含まれる情報収集モジュールは広告配信および利用統計のためにそれぞれモジュール提供元サーバと通信を行う

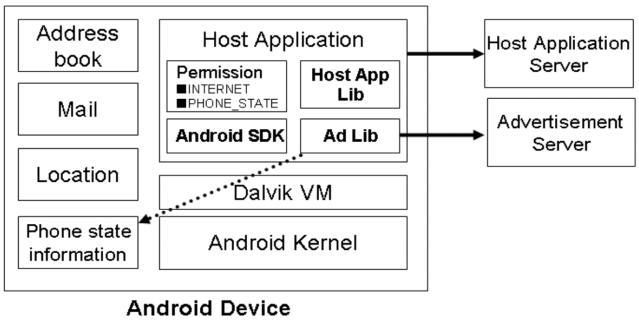

図 2 情報収集モジュールを含んだアプリケーションの概要．情報収集モジュールによる端末上の情報へのアクセスならびに外部への送信は，アプリケーションに付与されたパーミッションの範囲で行われる

2.2 情報収集モジュール

情報収集モジュールによる通信の概要を**図 1** に示す．代表的な情報収集モジュールとしては，ターゲット広告による広告配信を目的とし，利用者の端末上で取得した情報を用いる Google Inc. の AdMob[10]，また，アプリケーションの利用統計のために端末上の情報を用いる Flurry [11] が提供する情報収集モジュールがある．これらの情報収集モジュールは，アプリケーションにライブラリとして含まれ，アプリケーションの一部として動作する．そして，利用者によるアプリケーションの実行中に情報収集モジュール内部において，情報収集モジュール提供者が設置したサーバと通信を行う．提供される機能として，広告配信機能では，利用者端末への定期的な広告画像の取得と表示，利用統計機能では，アプリケーションの動作履歴や位置情報などの定期的な送信が行われる．

3. 情報収集モジュールの課題

情報収集モジュールを含んだアプリケーションの概要を**図 2** に示す．情報収集モジュールは，ライブラリとして配布され，アプリケーションに同梱される．そのため，アプリケーションにおいて情報収集モジュールを利用するには，情報収集モジュールの提供する機能が必要とするパーミッションをアプリケーションに付与しなければならない．その結果，利用者にアプリケーションに対する不信感を持た

表 1　HTTP 通信における拡張子毎の HTTP GET リクエスト 数

Content Type	# HTTP GET Request
jpg	13941
png	12539
gif	9197
php	5888
js	5114
css	2870
html	2093
json	841
txt	380
jpeg	282
JPG	128
Total	53273

表 2　HTTP GET リクエストにより取得した画像種別とサイズ毎のファイル数

Content Type	Image Size			
	320x48	320x50	300x48	300x50
jpg	57	278	2	4
png	271	870	0	59
gif	248	1197	0	115
jpeg	25	249	0	20
Total	601	2594	2	198

れる可能性がある．不信感を持たれる場合としては，アプリケーション本体の機能に必要なパーミッションと情報収集モジュールのパーミッションが組み合わさった場合やパーミッションを必要としないアプリケーションに情報収集モジュールのパーミッションが付与された場合などが想定される．

我々は，2012 年 1 月から 4 月の期間において，日本向け Google Play の Top 100 および新規更新に含まれたアプリケーションのうちから 1,188 個を抽出し，パーミッションの組み合わせを調査した．その結果，727 個のアプリケーション (61%) が **INTERNET** と端末上の情報にアクセスするパーミッションを要求していることが判明した [8]．これらのアプリケーションでは，利用者の同意を得ずに端末上の情報にアクセスし，その情報をネットワークを利用して外部に送信することが可能である．しかし，現在の Android パーミッションフレームワークでは，アプリケーションの一部やライブラリ単位でのパーミッションの制御は困難であり，情報収集モジュールなどによる端末上の情報送信が判明しても，利用者による動作制限はできない．

3.1　情報収集モジュールの通信内容

情報収集モジュールの通信内容を把握するために，アプリケーション 1,188 個より収集した HTTP 通信を調査した．調査環境は，Nexus S， Android 2.3.4 である．**表 1** に拡張子毎の HTTP GET リクエスト（以下，リクエスト）数を，**表 2** にリクエストにより取得した画像種別とサイズ毎のファイル数を示す．リクエストにおいては，画像ファイルに対するリクエストが最も多く，次に，PHP，JavaScript，HTML や CSS などのコンテンツであった．また，リクエストにより取得した画像ファイルにおいては，横 320 ピクセル，縦 50 ピクセル前後の画像が一定数あり，これらの多くは，情報収集モジュールによる広告配信にて取得された広告画像であると確認できた．

3.2　情報収集モジュールによる広告画像の取得手順

調査対象アプリケーションにおいて，情報収集モジュールによる広告画像の取得が確認できたことから，情報収集モジュール 120 個 [2], [3], [4], [5] がどの程度のアプリケーションに含まれて通信しているか調査し，通信を確認した情報収集モジュールについては，広告画像を取得するまでの一連の HTTP 通信について解析を行った．情報収集モジュールによる通信かの識別は，HTTP ヘッダの Host に含まれるドメイン名が情報収集モジュールの名称を含んでいるかにより行った．調査結果を**表 3** に示す．

調査結果より，多数のアプリケーションが複数の情報収集モジュールを含んでおり，HTTP 通信のドメイン名より 46 個の既知の情報収集モジュール（表 3 のドメイン数）による通信が含まれていることが判明した．また，1 つ以上の情報収集モジュールを含むアプリケーションは 797 個であった．情報収集モジュールとしては，“doubleclick.net” や “admob.com” と Google の情報収集モジュールが含まれたアプリケーションが多く，“adlantis.jp” や “mediba.jp” などの日本における情報収集モジュールを含んだアプリケーションも確認できた．また，1 つのアプリケーションにお

表 3 情報収集モジュールの通信が含まれたアプリケーション数，情報収集モジュールによる広告画像取得方法．各ドメインが情報収集モジュールを示す．多くのアプリケーションが複数の情報収集モジュールを含んでいるため，アプリケーション数は調査対象アプリケーション 1,188 個以上となっている

No.	Advertisement Module	# Application	Advertisement Path			
			HTML	JavaScript	HTTP Redirect	Other Domain
1	doubleclick.net	397	x			
2	google-analytics.com	351				x
3	admob.com	337		x		
4	googlesyndication.com	244				x
5	flurry.com	110				
6	adlantis.jp	78		x		
7	adwhirl.com	57	x		x	x
8	medibaad.com	49				x
9-11	inmobi.com, inmobicdn.net, inmobi-jp.com	44		x		x
12	mediba.jp	30		x		
13	mydas.mobi	24	x			
14	airpush.com	19				
15	mobclix.com	17		x		x
16	wiyun.com	14	x			
17	mopub.com	10	x			x
18	mobfox.com	10	x			x
19-46	Under 10 applications modules	76	14	9	1	9
Total	46	1875	20	14	2	17

いて複数の情報収集モジュールの通信が含まれている場合も存在することから，アプリケーションが用途や状況に応じて情報収集モジュールを使い分けていると考えられる．

情報収集モジュールによる広告画像の取得までの HTTP 通信としては，HTML，JavaScript，HTTP リダイレクトおよび他ドメインへの遷移などを経ており，一連の規則性があることが判明した．HTML や JavaScript を用いているのは，サーバ側での広告配信の動的制御を行うためだと考えられる．これにより，利用者単位で異なる広告画像の表示，広告画像の一定期間毎の変更といった任意のタイミングでの操作をアプリケーション内の情報収集モジュールを更新せずに行える．また，HTTP リダイレクトや他ドメインへの遷移は CDN (Contents Delivery Network) による負荷軽減や端末の動作している地域毎に最適な広告配信を行うためだと考えられる．

4. 提案手法

提案手法では，問題とされている情報収集モジュールにおいて，広告を配信するための一連の HTTP 通信が，HTML や JavaScript の取得から始まり，画像の取得で終了することに着目し，その動作を元にアプリケーションの HTTP 通信をグラフ化する．情報収集モジュールの HTTP 通信の検知は，情報収集モジュールの HTTP 通信のグラフとアプリケーションの HTTP 通信のグラフ間で，グラフの距離の計算を行い，情報収集モジュールと類似した HTTP 通信のグラフを抽出することで行う．

4.1 グラフの定義

提案手法で扱うグラフはラベル付きの頂点，ラベル付きの辺から構成される．L_V と L_E を頂点のラベル，辺のラベルの有限集合とする．グラフを $G = (V, E, \mu, \nu)$ の 4 項組みで表す．V は頂点 v_i の有限集合，$E \subseteq V \times V$ は辺 $e_{ij} = (v_i, v_j)$ の集合，$\mu : V \to L_V$ は頂点へのラベル割り当て関数，$\nu : E \to L_E$ は辺へのラベル割り当て関数とする．$V = \emptyset$ ならば，G は空グラフとする．また，$E_{i*} \subseteq E$ は頂点 v_i を端点とする辺の集合とする．

4.2 HTTP 通信のグラフ化

HTTP 通信より作成するグラフにおいて，頂点はクライアントからの HTTP リクエスト毎に割り当て，頂点へのラベルは HTTP の Request Line および Content Type とする．任意の HTTP リクエストに対応する頂点 v のラベルを $L_V(v) = (rline, ctype)$ により表す．

HTTP はステートレスなプロトコルであり，サーバとクライアント間での HTTP 通信は，クライアントからのリクエスト，サーバからのレスポンスによる 1 対 1 の組み合わせにより完結する．そこで，情報収集モジュールによる広告画像の取得に至るまでの HTTP リクエスト間の関係性を示す情報として，以下の情報を利用する．

- `HTTP Message-Body URL`：HTML や JavaScript のコンテンツに含まれる URL を参照し，別の HTTP 通信を開始．
- `HTTP Cookie`：複数の HTTP 通信でのセッションの

維持に Cookie を利用.

- `HTTP Persistent Connections`：1 つの TCP セッションにおける複数の HTTP 通信.
- `HTTP Redirect`：他の URL への HTTP 通信のリダイレクト.

これらの情報を HTTP 通信間の関係性として, HTTP リクエストの頂点間の辺および辺へのラベルとして割り当てる. 任意の 2 つの HTTP リクエストの頂点間 v_1,v_2 の辺 $e_{12} = (v_1, v_2)$ のラベルを $L_E(e_{12}) = (link, cookie, persistent)$ により表す.

4.3 グラフの距離

任意の 2 つのグラフの類似性は，グラフマッチングと呼ばれ多くの研究が行われている [12]．提案手法では，その中より，グラフの距離 [13] を用いる．これは, 2 つのグラフにおける最大共通部分グラフの大きさ利用し，グラフの類似性を計算する．2 つの空グラフでないグラフ G_1，G_2 の間の距離 d は，$d(G_1, G_2)$ で表し，以下の式で求められる．

$$d(G_1, G_2) = \frac{|mcs(G_1, G_2)|}{max(|G_1|, |G_2)|}$$

$mcs(G_1, G_2)$ はグラフ G_1，G_2 の最大共通部分グラフ，$|G|$ はグラフ G の頂点数，max は最大値を返す関数とする．

4.4 グラフ化した HTTP 通信間の距離の計算

4.2 節にもとづいて，グラフ化した HTTP 通信間の距離は 4.3 節のグラフの距離 d の計算により行う．グラフの距離の計算に用いる最大共通部分グラフは，バックトラックを使い最大共通部分グラフを求める手法 [14] を使用する．その手法を**図 3** に示す．提案手法では，グラフにおいて，HTTP 通信の特徴同士が一致した場合のみ同型グラフとするために，頂点と辺の集合のマッチングを行う．

図 3 のアルゴリズムは，2 つのグラフ化した HTTP 通信 G_1，G_2 を入力とし，G_2 から G_1 への置換行列 P を出力とする．

まず，G_1，G_2 の隣接行列を M_1，M_2 とし，G_1，G_2 の頂点数を n,m に，$P = (p_{i,j})$ を $n \times n$ の置換行列で $p_{ij} = 0;\ i, j = 1, \ldots, n$ とする．なお，探索のカウンタを $k = 1$ とする．つづいて，7 行目の関数 Backtrack にて，G_2 から G_1 への置換行列 P を探索する．$k > m$ ならば，探索を終了し，その時点の置換行列 P を返す．次に，12, 13 行目にて頂点 $v_{1,i}, v_{2,k}$ および辺の集合 $E_{1,i*}$，$E_{2,k*}$ のマッチングを 4.5 節のマッチング方法により行う．マッチングした場合，$p_{ki} = 1$ とし，i 行の $j \neq i$ 列は $p_{kj} = 0$ とする．

最後に，20 行目において，P が G_2 から G_1 への部分同型グラフとなる置換行列かの判定を行う．$S_{i,n}(P)$ は P に含まれる $i \times n$ の置換行列として，i 番目の頂点までにおける G_2 から G_1 への頂点の部分的な一致を示しており，$S_{i,i}(M_2) = S_{i,n}(P)M_1(S_{i,n}(P))^T$ が成り立つ場合，$S_{i,n}(P)$ は G_2 における i 番目の頂点までの部分グラフが G_1 の部分グラフの同型グラフであることを示す．そして，カウンタを更新し，関数 Backtrack を呼び出す．

Algorithm 1 Subgraph Isomorphism using Ullman's method

Input: $G_1 = (V_1, E_1, \mu_1, \nu_1), G_2 = (V_2, E_2, \mu_2, \nu_2)$
Output: Matrix P represents a subgraph isomorphism from G_2 to G_1

```
 1: n ⇐ |V1|
 2: m ⇐ |V2|
 3: M1, M2 are adjacency matrices of G1, G2
 4: P = (p_ij) is n × n initial permutation matrix
 5: k ⇐ 1
 6: Backtrack(M1, M2, P, k)
 7: function Backtrack (M1, M2, P, k)
 8: if k > m then
 9:     return P
10: end if
11: for all i ⇐ 1 to n do
12:     if Try to vertex label match (v_{1,i}, v_{2,k}) then
13:         if Try to edge label match (E_{1,i*}, E_{2,k*}) then
14:             p_ki ⇐ 1
15:             for all j ⇐ 1 to n do
16:                 if j ≠ i then
17:                     p_kj ⇐ 0
18:                 end if
19:             end for
20:             if S_{k,k}(M2) = S_{k,n}(P)M1(S_{k,n}(P))^T then
21:                 Backtrack(M1, M2, P, k + 1)
22:             end if
23:         end if
24:     end if
25: end for
26: end function
```

図 3 グラフ化した HTTP 通信間の共通部分グラフを求めるアルゴリズム

図 3 のアルゴリズムを用いて，任意の 2 つのグラフ化した HTTP 通信 G_1，G_2 において，最大共通部分グラフを求めるのにかかる時間計算量は，最良の場合 $O(|G_1||G_2|)$，最悪の場合 $O(|G_2|^2|G_1|)$ である．

4.5 頂点と辺の集合のマッチング

任意の 2 つのグラフ化した HTTP 通信 $G_1 = (V_1, E_1, \mu_1, \nu_1)$，$G_2 = (V_2, E_2, \mu_2, \nu_2)$ における頂点と辺の集合のマッチングは以下のように行う．

- 頂点のマッチング：頂点 $vi \in V_1$，$vj \in V_2$ のマッチングは，頂点のラベル $L_{V_1}(v_i) = (rline_i, ctype_i)$，$L_{V_2}(v_j) = (rline_j, ctype_j)$ を用い．$rline_i = rline_j$，$ctype_i = ctype_j$ あるいはどちらかが一致した場合，マッチングしているとする．
- 辺の集合のマッチング：辺の集合 $E_{i*} \subseteq E_1$，$E_{j*} \subseteq E_2$ のマッチングは, 各辺の集合に含まれる任意の 2 つの辺 $e_{ia} \in E_{i*}$，$e_{jb} \in E_{j*}$ において，辺のラベル $L_{E_1}(e_{ia})$，$L_{E_2}(e_{jb})$ が $L_{E_1}(e_{ia}) = L_{E_2}(e_{jb})$ の場合，マッチングしているとする．

5. 提案手法の評価

5.1 評価環境

提案手法の評価には，1,188 個のアプリケーションから収集したネットワークトラフィック（以下，トラフィック）を用いた．これらのアプリケーションは，2012 年 1 月から 4 月の期間において，日本向け Google Play Top 100 および新規更新に含まれたうちから選択した．トラフィックは，アプリケーションを Android 上で 5 分から 15 分動作させることで収集した．Android 端末には Nexus S， Android 2.3.6 を用いた．トラフィックの総量は，6,093,682 パケット，5.6 GBytes であり，トラフィックにおける HTTP 通信は 5,565,529 パケット，さらに HTTP GET/POST リクエストは 107,859 パケットである．

提案手法を評価するデータセットとして，まず，4.2 節のグラフ化手法により HTTP 通信のグラフ化を行う．グラフ化した HTTP 通信のうち情報収集モジュールのグラフは 4,696 個である．なお，本評価において，暗号化通信は用いない．

5.2 評価内容

提案手法の評価のために，データセットのうち情報収集モジュールのグラフより N 個のグラフをランダムに抽出する．N は，100，200 とした．

評価は，提案手法による，情報収集モジュールの HTTP 通信に対する検知精度を確認するために，N 個の情報収集モジュールのグラフを用い，N 個以外の情報収集モジュールのグラフに対し，どの程度，情報収集モジュールの HTTP 通信と判定したかにより行う．

5.3 評価結果

情報収集モジュールの HTTP 通信に対する提案手法によるグラフの距離毎におけるグラフ数および検知精度を**図 4** に示す．X 軸はグラフの距離，左側の Y 軸はグラフ数，右側の Y 軸は検知率である．情報収集モジュールの HTTP 通信のグラフ $4,696 - N$ 個に対し，$N = 100$ の場合，グラフの距離が 1，つまり情報収集モジュールの HTTP 通信と完全に一致したグラフ数は 1,499 個，検知率は 32% ，距離が 0.5，つまり情報収集モジュールの HTTP 通信のグラフが包含されているグラフ数は 2,641 個，検知率は 57%であった．また，$N = 200$ の場合，グラフの距離が 1 のグラフ数は 2,699 個，検知率は 58%，距離が 0.5 までのグラフ数は 3,343 個，検知率は 72%であった．

5.4 評価結果の考察

評価結果より，情報収集モジュールの HTTP 通信のグラフを一定数用いることで，他の情報収集モジュールの HTTP 通信を検知可能なことを示した．提案手法において

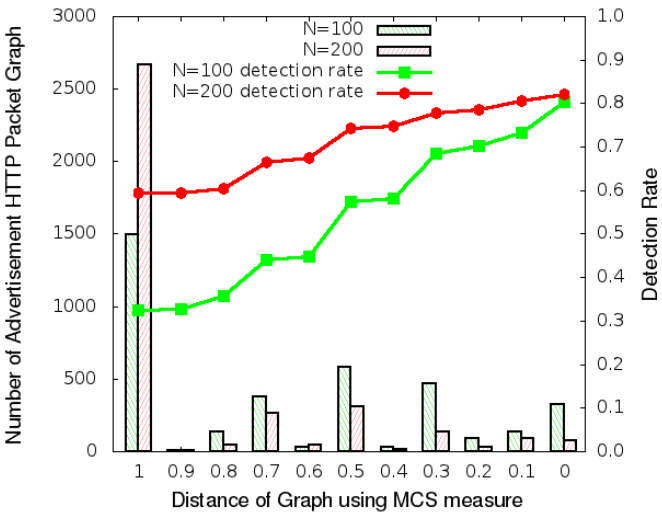

図 4 情報収集モジュールの HTTP 通信に対するグラフの距離毎のグラフ数および検知率

は，グラフの距離をしきい値として，情報収集モジュールの HTTP 通信か判定する必要があるため，どの値を基準とするかは難しい．しかし，グラフの距離が大きい場合ほど，情報収集モジュールの HTTP 通信であるといえる．評価結果では，$N = 200$ にて，グラフの距離が 1 に近い情報収集モジュールの HTTP 通信のグラフが多いため，より多数の情報収集モジュールの HTTP 通信パターンを網羅していると考えられる．

提案手法では，HTTP 通信に含まれる URL や送信先 IP アドレスの一致だけではなく，アプリケーションにより送受信された一連の HTTP 通信の関係性に着目しグラフ化を行い，情報収集モジュールの HTTP 通信かの判別している．これにより，情報収集モジュールのバージョンアップなどにより，その通信先が変化した場合でも，提案手法を用いることで，情報収集モジュールの HTTP 通信かどうか判別可能であるといえる．

6. 関連研究と議論

Android においては，多数のアプリケーションが利用者に関する情報を利用し，外部に送信する可能性があることがアプリケーションの静的解析や動的解析により指摘されている [2], [3]．アプリケーションに含まれる情報収集モジュールは，利用者が意図しないうちに情報漏えいに繋がる危険性があることから，情報収集モジュールに特化した解析が行われており，端末識別情報の送信内容の確認 [4]，100 個以上の情報収集モジュールの確認 [5]，情報収集モジュールが要求するパーミッション数の長期的な観測 [15] が行われている．

アプリケーションによる情報漏えいへの対策としては，パーミッションフレームワークの拡張によるアプリケーション動作の制限 [16]，Taint 解析による Android フレーム

ワークにおけるアプリケーションの情報フローの監視 [17], アプリケーションへの強制的なポリシー適用 [18], [19], [20] や情報収集モジュールのパーミッションのみ分離する手法 [6], [7] が提案されている．これらの手法はアプリケーションによる情報漏えいを防ぐ手法として有効であるが，Android フレームワークやアプリケーションの改良を必要としている．また，Android フレームワークの改良を要求しない手法としては，シグネチャを用いた情報漏えい検知手法 [8] や事前にアプリケーションの振る舞いを調査しておき，利用者へ提示する手法 [21] が提案されている．

アプリケーションにより送受信されるネットワークトラフィックを解析し，その特徴からプロトコル仕様を推測し，通信を把握する手法は，マルウェアによる攻撃やボットのトラフィック検出に用いられている [22], [23], [24], [25]．また，オートマトンによりボットの挙動やネットワークプロトコルをネットワークトラフィックからモデル化し，脆弱性を検出する手法 [26]，マルウェアの計算機上での振舞いとして情報フローとシステムコールをグラフ化し，効率的にマルウェアを検出する手法 [27] や Web 上の悪意を持った広告おける連続したリンクに着目しドメインの遷移系列としてモデル化し検出する手法 [28] が提案されている．

アプリケーションに含まれる情報収集モジュールに対する先行研究では，静的解析によるアプリケーションに含まれる情報収集モジュールの特定やコードからの挙動の解析，または，動的解析による端末上におけるアプリケーションの情報フローや送信情報に着目した解析が行われており，アプリケーションの HTTP 通信のみを利用した情報収集モジュールの検知は行われていない．提案手法においては，アプリケーションの HTTP 通信のみを情報収集モジュールの検知に利用しているため，Android フレームワークの改良や端末上でのアプリケーションの挙動に依存することなく適用可能である．さらに，情報収集モジュールが動作中に取得する HTML や JavaScript のデータなどアプリケーションの静的解析のみでは把握が困難な部分が存在するため，今後，提案手法により検知可能な情報収集モジュールの動作についてより詳細に解析していきたい．

提案手法では，情報収集モジュールが広告配信を行う一連の HTTP 通信に着目し，グラフ化を行っている．先行研究におけるオートマトンによるモデル化やネットワークトラフィックからのプロトコル仕様の詳細な解析と組み合わせることで，より正確に情報収集モジュールの HTTP 通信のモデル化を行える可能性がある．今後，情報収集モジュールの HTTP 通信のグラフ化の正確さの調査ならびに検知精度への影響を勘案し，より正確なグラフ化により検知精度の向上が可能かどうか検討したい．さらに，最大共通部分グラフを求める手法では，提案手法における手法 [14] よりも時間計算量の優れた計算手法 [29] が提案されており，提案手法の高速化についても検証を行いたい．

7. まとめ

情報収集モジュールが多くの無料アプリケーションに含まれており，広告サービスとして広く利用されている．しかし，情報収集モジュールによる利用者の意図しない情報漏えいの可能性が指摘されている．

我々は，アプリケーションの HTTP 通信を調査し，情報収集モジュールによる広告画像の取得が実際に行われていること，および広告画像の取得にかかる一連の HTTP 通信が HTML や JavaScript を利用して行われていることを示した．そして，提案手法として，アプリケーションに含まれる情報収集モジュールの HTTP 通信の関係性に着目することでグラフ化し，グラフの距離を用いて，アプリケーションに含まれる情報収集モジュールの HTTP 通信を検知する手法を提案し，その評価を行った．評価におけるデータセットとして，1,188 個のアプリケーションの HTTP 通信 5,565,529 パケットから提案手法を用いてグラフ化した情報収集モジュールの通信の 4,696 個を用いた．そのうち，200 個の情報収集モジュールのグラフによる評価の結果，提案手法は，情報収集モジュールの通信が包含されるグラフの距離が 0.5 までにおいて，72%の検知精度を示した．

謝辞

本研究の実施にあたり評価データ収集に協力いただいたセコム株式会社 IS 研究所 横島 浩氏に感謝いたします．

参考文献

[1] RSSPHONE: Google Play Store: 800,000 apps and overtake Apple AppStore!, RSSPHONE (online), available from ⟨http://www.rssphone.com/google-play-store-800000-apps-and-overtake-apple-appstore/⟩ (accessed 2013-04-25).

[2] Hornyack, P., Han, S., Jung, J., Schechter, S. and Wetherall, D.: These Aren't the Droids You're Looking For: Retrofitting Android to Protect Data from Imperious Applications, *18th ACM Conference on Computer and Communications Security (CCS 2011)* (2011).

[3] Enck, W., Octeau, D., McDaniel, P. and Chaudhuri, S.: A Study of Android Application Security, *20th USENIX Security Symposium 2011* (2011).

[4] Stevens, R., Gibler, C. and Crussell, J.: Investigating User Privacy in Android Ad Libraries, *Mobile Security Technologies (MoST 2012)* (2012).

[5] Grace, M., Zhow, W., Jiang, X. and Sadeghi, A.-R.: Unsafe Exposure Analysis of mobile In-App Advertisements, *5th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec 2012)* (2012).

[6] Pearce, P., Felt, A. P., Nunez, G. and Wagner, D.: AdDroid: Privilege Separation for Applications and Advertisers in Android, *7th ACM Conference on Computer and Communications Security (ASIACCS 2012)* (2012).

[7] Shekhar, S., Dietz, M. and Wallach, D. S.: AdSplit: Separating smartphone advertising from applications, *21st USENIX Security Symposium 2012* (2012).

[8] Kuzuno, H. and Tonami, S.: Signature Generation for Sensitive Information Leakage in Android Applications, *The 1st*

International Workshop on Mobile Data Analytics (MoDA 2013) (2013).

[9] Google: Android Developers Manifest.permission, Google (online), available from ⟨http://developer.android.com/reference/android/Manifest.permission.html⟩ (accessed 2013-04-30).

[10] Google: Admob, Google (online), available from ⟨http://www.google.com/ads/admob/⟩ (accessed 2013-04-30).

[11] Flurry: Flurry Analytics, Flurry (online), available from ⟨http://www.flurry.com/⟩ (accessed 2013-04-30).

[12] Conte, D., Foggia, P., Sansone, C. and Vento, M.: Thirty years of Graph Matching in Pattern Recognition, *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 18, No. 3, pp. 265–298 (2004).

[13] Bunke, H. and Shearer, K.: A graph distance metric based on the maximal common subgraph, *Pattern Recognition Letters 19* (1998).

[14] Ullman, J.: An algorithm for subgraph isomorphism, *Journal of the Association for Computing Machinery*, Vol. 23, No. 1, pp. 31–42 (1976).

[15] Book, T., Pridgen, A. and Wallach, D. S.: Longitudinal Analysis of Android Ad Library Permissions, *Mobile Security Technologies (MoST) 2013* (2013).

[16] Nauman, M., Khan, S., Alam, M. and Zhang, X.: Apex: Extending Android Permission Model and Enforcement with User-defined Runtime Constraints, *ACM Symposium on Information, Computer and Communications Security (ASIACCS 2009)* (2009).

[17] Enck, W., Gilbert, P., Chun, B.-G., Cox, L. P., Jung, J., McDaniel, P. and Sheth, A. N.: TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones, *9th USENIX Symposium on Operating Systems Design and Implementation (OSDI 2010)* (2010).

[18] Jeon, J., Micinski, K. K., Vaughan, J. A., Reddy, N., Zhu, Y., Foster, J. S. and Millstein, T.: Dr. Android and Mr. Hide: Fine-grained security policies on unmodified Android, *19th ACM Conference on Computer and Communications Security (CCS 2012)* (2012).

[19] Xu, R., Saidi, H. and Anderson, R.: Aurasium: Practical Policy Enforcement for Android Applications, *21st USENIX Security Symposium 2012* (2012).

[20] Hao, H., Singh, V. and Du, W.: On the Effectiveness of API-Level Access Control Using Bytecode Rewriting in Android, *8th ACM Conference on Computer and Communications Security (ASIACCS 2013)* (2013).

[21] Rosen, S., Qian, Z. and Mao, Z. M.: AppProfiler: A Flexible Method of Exposing Privacy-Related Behavior in Android Applications to End users, *Third ACM Conference on Data and Application Security and Privacy (CODASPY 2013)* (2013).

[22] Ingols, K., Lippmann, R. and Piwowarski, K.: Practical Attack Graph Generation for Network Defense, *The 22th Annual Computer Security Applications Conference (ACSAC 2006)* (2006).

[23] Wondracek, G., Comparetti, P. M., Kruegel, C. and Kirda, E.: Automatic Network Protocol Analysis, *Network and Distributed System Security Symposium (NDSS 2008)* (2008).

[24] Cui, W., jayanthkumar Kannan and Wang, H. J.: Discoverer: Automatic Protocol Reverse Engineering from Network Traces, *16th USENIX Security Symposium 2007* (2007).

[25] Lin, Z., Jiang, X., Xu, D. and Zhang, X.: Automatic Protocol Format Reverse Engineering through Context-Aware Monitored Execution, *Network and Distributed System Security Symposium (NDSS 2008)* (2008).

[26] Comparetti, P. M., Wondracek, G., Kruegel, C. and Kirda, E.: Prospex: Protocol Specification Extraction, *IEEE Security and Privacy (S&P 2009)* (2009).

[27] Kolbitsch, C., Comparetti, P. M., Kruegel, C., Kirda, E., Zhow, X. and Wang, X.: Effective and Efficient Malware Detection at the End Host, *18th USENIX Security Symposium 2009* (2009).

[28] Li, Z., Zhang, K., Xie, Y., Yu, F. and Wang, X.: Knowing Your Enemy: Understanding and Detecting Malicious Web Advertising, *19th ACM Conference on Computer and Communications Security (CCS 2012)* (2012).

[29] Messmer, B. T. and Bunke, H.: Subgraph Isomorphism in Polynomial Time, Technical report, Technical Report IAM-95-003, University of Bern (1995).