個人情報保護を目的としたフレームワークの提案

松永 崇秀 †　　高橋 健一 †　　川村 尚生 †　　菅原 一孔 †

† 鳥取大学大学院 工学研究科

あらまし　近年，インターネットの普及に伴い，オンラインショップや施設の予約など様々なネットワークサービスが利用されている．これらのサービスのいくつかは利用者に対して個人情報の提供を要求する．しかし，利用者は提供した個人情報が実際にどのように利用されるか知ることができないため，サービス提供者に個人情報を提供することに不安を感じる．そこで，利用者が個人情報の利用方法を指定することができる仕組みを提案する．サービス提供者は自身の持つプログラムで個人情報を処理する．そこで，このプログラムに利用者が指定した利用方法を反映させることで，安心してネットワークサービスを利用できるフレームワークを実現する．

A Framework for Protecting Personal Information

Takahide Matsunaga†　　Kenichi Takahashi†　　Takao Kawamura†
Kazunori Sugahara†

†Graduate School of Engineering, Tottori University

Abstract　Nowadays, various network service, such as online shops and reservation of facilities, have been used with the spread of the Internet. Some of these services request to offer personal information to users. However, we cannot know how offered personal information is used. Thus, we feels uneasy to offer personal information to service providers. In this paper, we propose a framework that an user can designate usage procedures of his/her personal information. Here, a service provider is processing user's personal information in its own program. Therefore, we realize a framework that enables to apply usage procedures designated from the user to the program of the service provider.

1　はじめに

近年，インターネットの普及に伴い様々なネットワークサービスが利用されている．例えば，Amazon や楽天などのオンラインショップやホテルの予約，オンラインバンキングなどが挙げられる．これらのサービスは，利用者に対して名前や住所，電話番号やクレジットカード番号などの個人情報の提供を求める．利用者は要求に従い，自身の個人情報を提供することでサービスを利用する．

しかし，近年では，情報漏洩事件 [1] やフィッシングサイトなどによる被害 [2]，サービス提供者による情報の不正利用などが多発している．これらの問題に対して，共通鍵暗号方式や公開鍵暗号方式などの暗号化技術や電子証明書，SSL や TLS のようなプロトコルなど，さまざまなセキュリティ技術が開発されてきた．しかし，これらの技術を利用するか否かはサービス提供者が決定するため，利用者にはその決定権がない．また，これらの技術を利用したとしても，利用者は一度渡した個人情報が実際にどのように利用されているか確認することができない．この

ため，利用者はサービス提供者に個人情報を提供することに不安を感じたとしても，個人情報を提供してサービスを利用するか，個人情報を提供せずにサービスを利用しないかという選択しかすることができない．

そこで，我々は利用者自身が個人情報の処理方法を決めることができる仕組みを提案する．一般的に，利用者の個人情報はサービス提供者が持つプログラムで処理される．そこで，提案モデルではこのプログラムの処理方法を利用者が指定した方法に書き換える．これにより，利用者は自身が指定した方法で個人情報の処理を行わせることができるようになる．この手法により，利用者が安心して個人情報を提供することが出来る仕組みを実現する．

以降，2 章でネットワークサービスの問題点について述べ，3 章で我々が提案するフレームワークについて述べる．4 章で本フレームワークで必要となるプログラムの変換について述べ，5 章でその実装について述べる．

2　ネットワークサービスの問題点

現在のネットワークサービスでは，利用者はサービス提供者の定める方法に従って個人情報を提供する．例えば，Amazon や楽天などのオンラインショップであれば，初回登録時は住所や氏名，電話番号などを，ログイン時は ID やパスワードなどを提供する必要がある．しかし，個人情報の利用方法はサービス提供者に委ねられている．すなわち，利用者は実際に提供した個人情報がどのように扱われているか知ることができず，一度提供した個人情報を保護することもできない．このため，利用者はサービス提供者がこれらの個人情報を不正利用しないと信頼していることが前提として必要となる．一方，近年では情報の漏洩や不正利用，フィッシングサイトなどによる被害が多発している．これにより，利用者はサービス提供者に個人情報を提供することや使用しているサイトの安全性に不安を感じる．このことから，利用者はサービス提供者のことを信頼できないということが考えられる．これを解決するためには，

【問題 1】 利用者にはサービス提供者による個人情報の利用方法がわからない

【問題 2】 一度提供した個人情報を利用者自身が保護する手段がない

に対処するための方法が必要となる．

また，個人情報の利用には漏洩などのリスクを伴う．このため，サービス提供者にとって，個人情報の利用や管理に関するすべての責任を負うことが負担となっている．実際に，個人情報の管理を代行サービス [4][5] に委託している企業も存在する．このため，

【問題 3】 個人情報を扱う上でのサービス提供者にかかる負担が大きい

への対処法も必要となる．

3　個人情報保護フレームワーク

現在のインターネットサービスでは，利用者が提供した個人情報はサービス提供者の持つ個人情報処理プログラムで処理されている（図 1）．しかし，利用者にはこのプログラムの処理内容がわからないため，【問題 1】，【問題 2】が発生する．

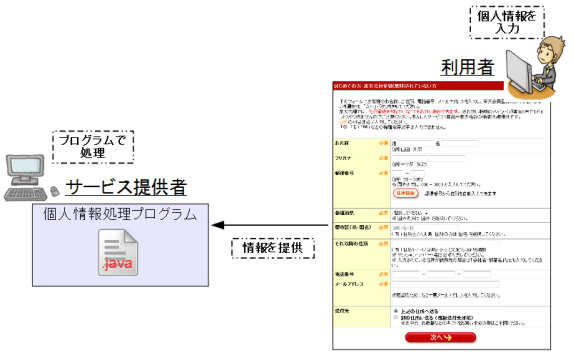

図 1: インターネットサービス利用の流れ

ここで，もしプログラムの処理部分を利用者が確認することができれば，処理内容を知ることができるため，【問題 1】の発生を軽減できる．さらに，利用者が自身の納得できる方法で個人情報の処理を行わせることができれば，【問題 2】も解決できる．また，処理方法を利用者が決定

するため,【問題 3】のサービス提供者の負担についても軽減することができる．

そこで，サービス提供者の持つプログラムを変換し，個人情報の処理を利用者が制御することが可能となる個人情報保護フレームワークを提案する．

3.1 概要

本フレームワークでは，サービス提供者の持つ個人情報処理プログラムに利用者が指定した処理方法を適用することで，利用者の意図を個人情報の処理に反映させる．これを実現するためには，利用者は自身の納得できる処理方法をサービス提供者に伝える必要がある．

ここで，サービス提供者による個人情報の処理方法はプログラムによって既に決まっている．このため，利用者が納得できる処理方法は，このプログラムに適用可能な処理方法である必要がある．そこで，プログラム中での情報の利用方法を示した利用ポリシを定義する．利用ポリシにより，利用者は間接的にサービス提供者による個人情報の利用方法を知ることができる．

また，個人情報の処理方法およびプログラムの変換方法を定義した保護ポリシを準備する．利用者は利用ポリシを参照することで，そのプログラムに適用可能である保護ポリシを選択し，サービス提供者に伝える．サービス提供者は保護ポリシに従ってプログラムを変換し，変換後のプログラムで個人情報の処理を行う．これにより，利用者が個人情報の保護方法を決定する．

ここで，一般の利用者にはプログラムに関する知識はないため，保護方法を自身で定義することは難しい．また，信頼出来ない保護ポリシを使用して変換を行った場合，変換後のプログラムがサービス提供者の目的とは違う動作をする（例えば，マルウェアの機能を持つように変換されるなど）危険性がある．このため，保護ポリシは信頼できる第三者機関 (TTP：Trusted Third Party) が設置した保護ポリシデータベースで管理されるものとする．TTP が確認した保護ポリシのみを利用することで，このような危険性を排除できるものと考える．また，プログラムが正しく変換されることを保証するために，変換も TTP で行う．個人情報保護フレームワークの概要を図 2 に示す．本フレームワークの動作の流れについては，紙面の都合で省略する．

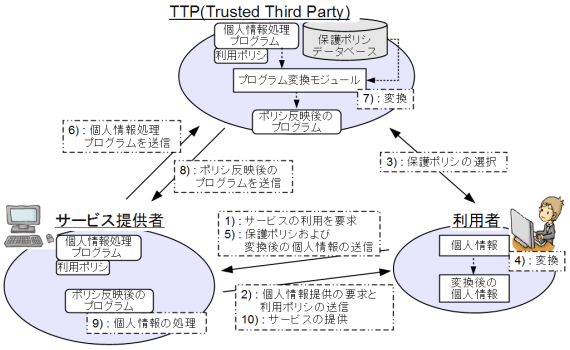

図 2: 個人情報保護フレームワークの概要

3.2 利用ポリシ

利用ポリシは，

- プログラムでの個人情報の利用方法の確認
- 適用可能な保護ポリシの選択

を可能とするために準備する．また，利用ポリシは，パーサによる詳細な解析を可能とするために XML 形式で定義する．利用ポリシの例を図 3 に示す．

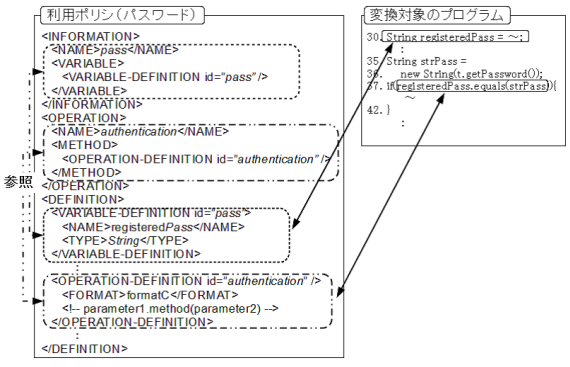

図 3: 利用ポリシの例

利用ポリシは，プログラム中で利用する情報 1 つにつき 1 つ用意する．利用ポリシ中の<INFORMATION>内の<NAME>には，サービス提供者による情報の識別名を定義する．この識別名は

Web ページの入力フォームに対応した値が指定される．このため，図 3 の例は，pass という値が割り当てられた入力フォームへ入力された情報に対する利用ポリシであることを表す．

<VARIABLE>には，プログラム中で対象の情報を扱うための変数を定義する．図 3 の例では，<DEFINITION>内の id="pass"である箇所を参照することにより，パスワードが String 型の変数 registeredPass として扱われていることがわかる．

<OPERATION>には，対象の情報のプログラム中での利用方法を定義する．図 3 の例では，パスワードが parameter1.equals(parameter2) という形式で使用されていることがわかる．他にも parameter1 および parameter2 に関する定義が存在するが，紙面の都合で省略している．

利用者は，利用ポリシを見ることでどの情報がどのように処理されているか知ることができるため，プログラムに適用可能な保護ポリシを選択することができる．また，プログラム中で変換対象の情報がどの変数に格納されているか伝えることができる．

3.3 保護ポリシ

保護ポリシは

- 利用者による個人情報の保護方法の選択
- 個人情報の保護方法（プログラムの変換方法）の定義

を可能とするために準備する．保護ポリシも利用ポリシと同様に，XML 形式で定義する．保護ポリシの例を図 4 に示す．

保護ポリシ中の<EXPLANATION>を見ることにより，その保護ポリシが情報をどのように保護するか知ることができる．また，<INFORMATION>には変換対象の情報が定義され，<OPERATION>にはその情報に対して行う操作が定義される．このことから，図 4 の例は「データ (password) をハッシュ変換し，変換したデータを用いて認証 (authentication) を行う」ための保護ポリシであることがわかる．

保護ポリシ

```
<EXPLANATION>
    データをハッシュ変換し，
    変換したデータを用いて認証を行う
</EXPLANATION>
<INFORMATION>password</INFORMATION>
<OPERATION id="authentication">
  authentication
</OPERATION>
<CONVERT-RULE>
  <DATA-CONVERT-RULE id="cast">
    hashPass ← hash(password, key)
  </DATA-CONVERT-RULE>
        :
  <OPERATION-CONVERT-RULE>
    authentication(hashPass, receivedPass)
      ← authentication(password, receivedPass)
  </OPERATION-CONVERT-RULE>
        :
  <DATA-DISCARD-RULE>
    password : forbidden
  </DATA-DISCARD-RULE>
</CONVERT-RULE>
```

図 4: 保護ポリシの例

<INFORMATION>や<OPERATION>により保護ポリシを検索することで，保護ポリシデータベースから目的の情報，操作を持つ保護ポリシを絞り込むことができる．さらに，絞り込んだポリシの<EXPLANATION>を見ることにより，利用者は自身の安心できる保護方法を選択することができる．

また，プログラムの変換方法はプログラム変換ルール (<CONVERT-RULE>) として定義する．プログラム変換ルールは複数のルールから構成され，データ変換，操作変換，データ破棄のいずれかのルールとなる．

データ変換ルール：<DATA-CONVERT-RULE>
データを生成するためのルールを定義する．例えば，暗号化によってデータを保護する場合，暗号化されたデータを生成する必要がある．このようにデータを生成するためのルールを「変換後の情報 ← 変換前の情報を変換」として定義する．図 4 の例では，パスワードをハッシュ変換し，変換したものを hashPass とするルールを示している．

操作変換ルール：<OPERATION-CONVERT-RULE>
変換前の情報に対して行われていた処理を，変換後の情報に対してそのまま行うことはできない．そこで，操作を変換するためのルールを定義する．操作変換ルールは「変

換後の操作 ← 変換前の操作」として定義する．図 4 の例では，パスワードと利用者から受け取った情報に対して行っていた操作を，hashPass と利用者から受け取った情報に対して行う操作に変換するルールを示している．

データ破棄ルール：<DATA-DISCARD-RULE>
変換前の情報の利用を禁止するためのルールを定義する．データ破棄ルールは「変換前の情報:forbidden」として定義する．図 4 では，元のパスワードに関する処理を禁止する例を示している．

これらのルールを記述した保護ポリシを準備することで，個人情報の保護方法を定義する．

4 プログラムの変換

サービス提供者の持つ個人情報処理プログラムは，保護ポリシと利用ポリシに従いプログラム変換モジュールによって変換する．変換は信頼できる第三者機関で行う．図 5 にプログラム変換モジュールによるプログラム変換の流れを示す．

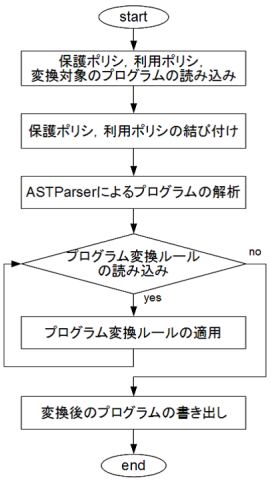

図 5: プログラム変換の流れ

プログラム変換モジュールはまず，保護ポリシと利用ポリシ，および変換対象の個人情報処理プログラムの読み込みを行う．次に，保護ポリシと利用ポリシの結びつけを行う．保護ポリシには個人情報の保護方法が定義され，利用ポリシには変換対象のプログラム中での個人情報の処理内容が定義されている．そのため，保護ポリシと利用ポリシを結びつけることで，プログラム中のどの処理 (変数) をどのように変換すればよいか判断できる．その後，プログラムの解析を行う．解析後，結びつけを行った変換ルールを変換対象のプログラムに適用し，プログラムを変換する．すべての変換ルールの適用後，プログラムを書き出す．これにより，利用者が安心できる処理方法が適用された個人情報処理プログラムを生成する．

4.1 ポリシの結びつけ

保護ポリシには保護対象の情報の変換方法が記されている．しかし，保護ポリシだけではサービス提供者の持つプログラム内のどの変数に保護対象の情報が格納されているかわからない．そこで，プログラム中での保護対象の情報の処理方法が記されている利用ポリシと結びつけを行う．図 6 に保護ポリシと利用ポリシの結びつけ例を示す．

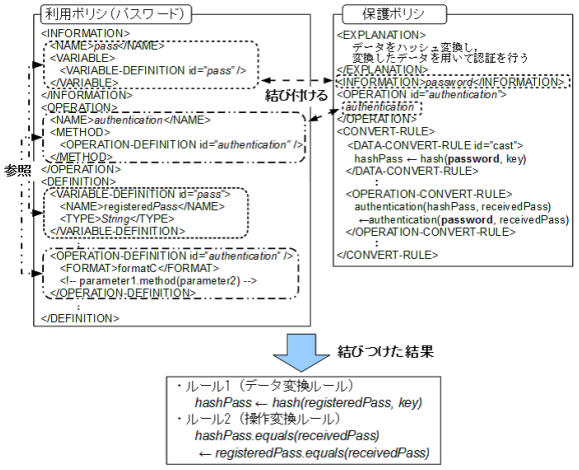

図 6: ポリシ構成例と結びつけ例

ポリシの結びつけは各ポリシ内のタグおよびその内容を解析，検索することで行う．図 6 では，<INFORMATION>内の<NAME>が pass となっているパスワードの利用ポリシに対して，<INFORMATION>が password となっている保護ポリシを利用者が選択した例を示している．

利用ポリシの<INFORMATION>内の<VARIABLE>と<DEFINITION>内の対応する箇所を参照することにより，サービス提供者の持つ個人情報処理プログラムでpassはString型の変数registeredPassに格納されていることがわかる．さらに，この保護ポリシはこの利用ポリシに対して選択されたものであるため，保護ポリシ中のpasswordは，プログラム中の変数registeredPassに対応することがわかる．

同様に，保護ポリシと利用ポリシの<OPERATION>を結びつけることで，保護ポリシで定義された操作をプログラム中のどの操作に対して適用すればよいか知ることができる．

このようにポリシの結びつけを行うことによって，保護ポリシが保護対象とする情報とその操作が，プログラム中でどの変数に格納され，どのような操作で処理されるか知ることができる．

4.2 プログラムの解析

プログラム中の変換対象の変数を特定するために，プログラムの解析を行う．本フレームワークでは，Java 言語で記述された個人情報処理プログラムを対象に変換を行うことを想定する．そのため，Java のソースコードを解析することができる ASTParser を利用する．ASTParser は EclipseJDT が提供する API の 1 つで，Java 言語のソースコードを解析し，抽象構文木 (AST:Abstract Syntax Tree) を生成する．抽象構文木とは，ソースコードからコメント文や空行などの実行する際に不要な情報を取り除いたデータ構造のことである．ASTParser により構文解析され，プログラムの各行は変数の宣言文や代入文，ループ文などのように意味付けされる．またその行はさらに字句へと分解され，変数名やメソッド名などのように意味付けされる．これにより，プログラム中でどの変数にどの情報を格納するか，どのメソッドでどの変数を使用するかなどを解析する．

4.3 プログラム変換ルールの適用

プログラムの解析後，プログラム変換ルールを適用する．プログラム変換ルールは複数のルールから構成されており，ルールによっては他のルールにより生成された情報を使用しなければ適用できないものが存在する．例えば，操作変換ルール内でデータ変換ルールにより生成される変数を使用している場合，操作変換ルールを先に適用することはできない．そこで，まずプログラム変換ルールの中から適用可能であるルールを抽出する．次に，ASTParser で解析した結果を元にルールを適用する箇所を検索し，そのルールを適用する．AST 中の変換対象の検索例および図 6 のルール 2（操作変換ルール）の適用例を図 7 に示す．

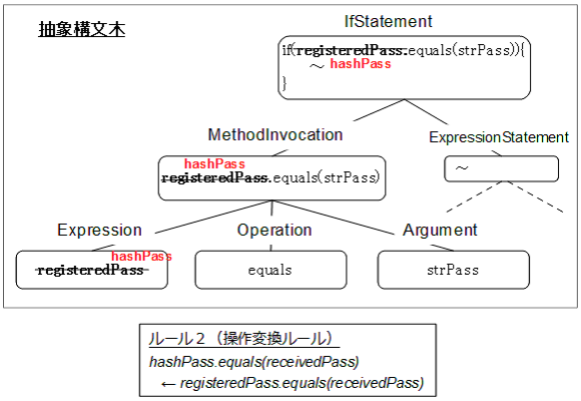

図 7: ノードの検索とポリシ適用例

図 7 は，registeredPass を hashPass に変換するルールを適用した例を示している．この例では，まず if 文が条件式とその内容に解析され，さらに条件式は registeredPass，equals，strPass に分解される．この時，探索により変換対象の registeredPass が格納されているノードを見つけることができる．そこで，このノードを操作変換ルールに従って hashPass に書き換える．これにより，registeredPass を hashPass に置き換えた AST を作成することができる．

これを繰り返し，すべてのルールを適用することで，プログラムを変換する．

4.4 プログラムの書き出し

変換後の AST からプログラムを再構成し，プログラムを書き出す．ハッシュ変換によりパスワードを保護し，変換後の情報で認証ができるようにプログラムを変換した例を図 8 に示す．

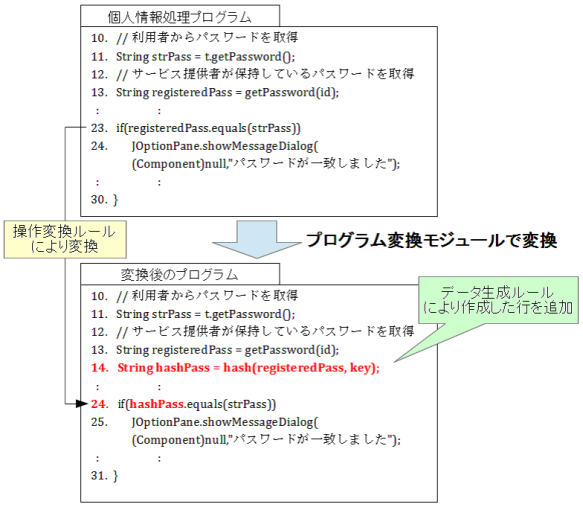

図 8: プログラムの変換例

4.3 節で示した変換例により，プログラムは図 8 の変換後のプログラムの 24 行目のように変換される．また，図 6 のルール 1（データ変換ルール) により，変換後のプログラムの 14 行目のようなコードが追加される．

すべてのルールを適用後，変換後のプログラムを Java 形式で書き出す．これをサービス提供者が実行することで，利用者が指定した処理方法にプログラムを変換することが可能となる．

5 実装

プログラム変換モジュールおよびユーザインターフェースを作成し，その実装を行った．図 9 に実装したユーザインターフェースを示す．

図 9: UI のポリシ選択画面

図 9 の例では，サービス利用時に ID とパスワードを要求するサービスを想定している．ユーザインターフェースでは，利用ポリシからサービス提供者が必要とする個人情報を解析することで，保護ポリシを適用可能である情報を表示 (図 9 左上のフレーム) し，また，各情報に適用可能な保護ポリシの一覧を表示 (図 9 左下) する．保護ポリシを選択することで，選択した保護ポリシの内容が表示される (図 9 右下)．利用者はこれを確認することで，各情報に対して適用する保護ポリシを選択する．最後に利用者が個人情報を入力すると，その情報が保護ポリシに従って変換され，サービス提供者に送信される．

変換された情報を受け取ったサービス提供者は，選択されたポリシに従ってプログラムを変換する．その後，変換後のプログラムで個人情報の処理を行う．

ここでは，生のパスワードを受け取り，パスワード認証に成功したか否かを表示するだけのサービスを想定して実験を行った．実験結果を図 10 に示す．

図 10: パスワードの認証結果

本実験では，利用者はハッシュ変換によりパスワードを保護する保護ポリシを選択している．このため，サービス提供者はハッシュ変換されたパスワードである「54cdc ･･･ 」を受け取り，保護ポリシによって変換された個人情報保護プログラムでパスワードの処理を行っている．この結果，正しく認証を行えることが確認できた．

6 関連研究

提供した個人情報がインターネットサービスでどのように利用されているか確認する手段として，プライバシーポリシー [3] を閲覧するという方法がある．プライバシーポリシーは多くの

サービス提供者が策定しており，収集する個人情報の利用目的や利用方法などが記されている．利用者はこれを見ることにより，サービス提供者による個人情報の利用方法を知ることができる．しかし，プライバシーポリシーは記述してある通りに個人情報が利用されることを保証しない．実際に，プライバシーポリシーを定義しているサイトでも情報漏洩事件が起きている．

また，収集する個人情報の利用方法を利用者に提示するフォーマットとして P3P(Platform for Privacy Preferences)[6] がある．P3P は利用者があらかじめ定めた個人情報の利用基準と各サイトのプライバシーポリシーを比較し，自動的に情報提供の可否を判断する．しかし，P3P もサイトがプライバシーポリシー通りに行動することを保証しない．

利用者が安心してサービスを利用するための仕組みとして，PPM（Privacy Policy Manager）[7] がある．PPM では，パーソナルデータの取り扱いに関するユーザプリファレンスを管理することによってデータの流通を制御する．同様に，開示する情報の粒度を制御することでプライバシを保護する研究として [8] がある．これらにより，利用者の要望にあった情報のみを送信し，その情報のみで利用できる範囲のサービスを利用することが可能となる．しかし，その情報単体で利用者の特定に繋がる情報は守ることができない．また，サービス提供者が必要とする情報も遮断することが可能であるため，サービスの利用に支障を来す可能性がある．

また，個人情報を送信しないことにより，悪意のあるアプリケーションから個人情報を守る研究 [9] がある．この研究では，個人情報の代わりに，アプリケーションルールにより生成した制御コマンドを送信することで情報を保護している．しかし，アプリケーションルールはサービス提供者が作成しているため，利用者が制御方法を指定することはできない．

7 おわりに

本稿では，インターネットサービスの利用時における個人情報提供への利用者の不安を改善するための手法として，個人情報保護フレームワークを提案した．個人情報保護フレームワークは，サービス提供者の持つ個人情報処理プログラムでの処理を利用者が指定した処理方法に変換し，変換後のプログラムで個人情報の処理を行う．これにより，利用者は自身の納得できる処理方法で処理を行わせることができるため，安心して情報を提供し，サービスを利用できる．

今後の課題として，様々なプログラムを変換可能とするために，プログラム変換モジュールやポリシを改良することが挙げられる．

参考文献

[1] Security NEXT．”情報漏洩事件・事故一覧”．http://www.security-next.com/category/cat191/cat25

[2] フィッシング対策協議会．”報告書”．https://www.antiphishing.jp

[3] IBM．”プライバシー・ポリシーの定義”．https://publib.boulder.ibm.com/tividd/td/ITPME/SC23-1284-00/ja_JA/HTML/p12plmst22.htm

[4] 顧客 BANK.com．”顧客データ管理”．http://www.kokyakubank.com/index.html

[5] JUMBO．”顧客データ管理代行”．https://www.jmb.co.jp/gyomu/Gyomu_CRM.html

[6] W3C．”Platform for Privacy Preferences Project”．http://www.w3.org/P3P/

[7] 中村徹ほか．”パーソナルデータ流通基盤: Privacy Policy Manager (PPM) の受容性評価”．SCIS2014，3D3-2，2014.

[8] 宮本崇弘ほか．”プライバシとサービス品質のトレードオフを考慮した個人情報制御機構の提案”．DEWS2005，6-A-01，2005.

[9] 田丸修平ほか．”プライバシを考慮したパーソナライゼーションを実現するアプリケーションフレームワーク”．情処研報，OS-93，p.49-56，2003.