L-016

個人情報保護を目的としたプログラム変換方法の検討
Program Modification Toward User Responsible Privacy

工藤 邦晃 † Kuniaki Kuto　**高橋 健一** † Kenichi Takahashi　**川村 尚生** † Takao Kawamura　**菅原 一孔** † Kazunori Sugahara

1　はじめに

現在，様々なサービスがインターネット上で提供される．例えば，ショッピングサイトやホテルの予約などが挙げられる．これらのサービスは，個人情報の提供を要求する．しかし，一旦渡した情報がどのように利用されるか利用者は確かめることができない．また，情報漏洩事件 [1] や不正利用，フィッシングサイトによる被害なども多発している．その結果，利用者は情報の不正利用や情報漏洩事件に苦しまなければならない．このことを避けるためには，利用者がサービス提供者に個人情報を渡さなければよい．しかし，利用者はサービスを利用することができなくなる．そのため，利用者の個人情報が悪用されないように，情報を保護するための仕組みが必要となる．そこで，利用者が個人情報の処理方法を指定可能なモデルを提案する．

本モデルでは，利用者が個人情報の処理方法を指定する．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．そのため，本モデルではサービス提供者が個人情報を処理するプログラムを利用者によって指定された処理方法に書き換える．サービス提供者は利用者が指定した処理方法に書き換えられたプログラムを通して個人情報を処理する．個人情報は利用者が選択した処理方法に従って処理されるため，利用者はサービス提供者による個人情報の悪用を妨げることができる．その結果，利用者は安心して自身の個人情報をサービス提供者に提供できる．

そこで，本稿では，本システムを実現するために，利用者によって選択された処理方法をサービス提供者が持つプログラムに適用するためのプログラムを変換方法について検討する．

2　提案モデル

本提案モデルは，サービスを利用する利用者とサービスを提供するサービス提供者からなる．利用者は自身が安心できる個人情報の処理方法をサービス提供者に伝え，サービス提供者にその処理方法で個人情報を処理させることで個人情報を保護する．そのため，利用者はサービス提供者に自身が安心できる処理方法を伝える必要がある．そこで，保護ポリシを導入する．利用者は保護ポリシを定義し，サービス提供者に伝えることで自身が安心できる個人情報の処理方法を伝えることができる．また，サービス提供者は自身が持つ個人情報処理プログラムに利用者が指定した保護ポリシを適用する必要がある．これは，変換モジュールによって行う．サービス提供者は変換モジュールによって個人情報処理プログラムを保護ポリシに定義された処理に変換する．しかし，保護ポリシだけではプログラム中で情報がどのように処理されているかわからず，変換モジュールでプログラムを変換することはできない．そこで，これらの事を定義した利用ポリシを導入する．利用ポリシを見ることでプログラムでの個人情報の処理方法，個人情報が格納される変数名を知ることができ変換できる．

保護ポリシ

保護ポリシは，ポリシの説明とプログラム変換ルールからなる．説明は自然言語で書かれ，利用者は説明を読むことで安心だと思うポリシを選択することができる．プログラム変換ルールはプログラムを説明文で書かれた処理方法に変換するためのルールである．説明文によって利用者は自身が安心できる処理方法 (保護ポリシ) を指定でき，プログラム変換ルールによって，その処理方法をサービス提供者のプログラムに適用することができる．

利用ポリシ

プログラムを変換するためには，サービス提供者が持つプログラムがどのような処理をどのように行うかを知る必要がある．そこで，プログラム内でどのように利用者の個人情報を利用するかを利用ポリシとして定義する．利用ポリシはサービス提供者が利用者に要求する個人情報ごとに準備され，利用ポリシにより，そのプログラムが個人情報をどのように処理するかを知ることができる．

プログラムへの利用者が指定した処理方法の適用はプログラム変換モジュールによって行われる．変換モジュールではサービス提供者のプログラムを保護ポリシと利用ポリシに従って変換することで，利用者が指定した処理方法を反映したプログラムを生成する．図 1 にプログラム変換モデルの概要を示す．

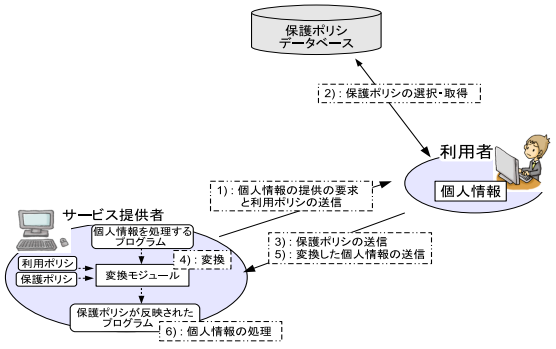

図 1: プログラム変換モデル

†鳥取大学大学院 工学研究科 情報エレクトロニクス専攻

1. 利用者はサービス提供者にサービスを要求する．この時，サービス提供者はサービスを利用するための条件として利用者に対して個人情報の提供を要求する．しかし，利用者はサービス提供者を信頼できず，個人情報を与えることに不安に感じる．そこで，利用者はサービス提供者による個人情報の処理方法を確認するために利用ポリシを要求する．サービス提供者はその個人情報に対する利用ポリシを利用者に送信する．
2. 利用者は利用ポリシにより，サービス提供者がどのように個人情報を処理しようとしているか知ることができる．そこで，利用者は保護ポリシデータベースにアクセスし，利用ポリシに定義された処理方法に合致する保護ポリシの一覧を得る．利用者はその中から個人情報の処理方法が定義された保護ポリシを取得する．
3. 利用者は指定した保護ポリシをサービス提供者に送信する．
4. サービス提供者は保護ポリシ，利用ポリシにより，個人情報処理プログラムを変換モジュールで保護ポリシが反映されたプログラムに変換する．
5. 利用者は保護ポリシに従い保護したい個人情報を変換し，変換した個人情報をサービス提供者に送信する．
6. サービス提供者は，変換された個人情報を保護ポリシが反映されたプログラムで処理する．

これにより，利用者は自身が選択した処理方法でサービス提供者に個人情報を処理してもらうことが可能となる．第 3 章で保護ポリシ，第 4 章で利用ポリシ，第 5 章でプログラムの変換方法について述べる．

3 保護ポリシ

利用者は保護ポリシにより，自分が安心できる個人情報の利用方法を選択する．利用者が保護ポリシを選択するためには，保護ポリシがどのような情報をどのような操作に対しどのように守るのかを理解できる必要がある．このため，保護対象となる情報を<INFORMATION>，保護対象となる操作を<OPERATION>，保護方法を<EXPLANATION>として定義する．<INFORMATION>や<OPERATION>を確認することでサービス提供者の持つプログラムに適用可能な保護ポリシの候補を絞り込むことができ，<EXPLANATION>を確認することで利用者が安心できる処理方法を選ぶことができる．

また，<INFORMATION>を<EXPLANATION>の方法で処理できるようにプログラムを変換するための方法は<CONVERT-RULE>で定義される．<CONVERT-RULE>はプログラムを変換するための複数のルールからなり，それらのルールに従ってプログラムを書き換えることで<EXPLANATION>の方法で<INFORMATION>を処理するようにプログラムを変換できる．保護ポリシの構成を図 2 に示す．

```
<EXPLANATION>
    自然言語で書かれた説明文
</EXPLANATION>
<INFORMATION> information </INFORMATION>
<OPERATION> operation </OPERATION>
<CONVERT-RULE>
    rules
</CONVERT-RULE>
```

図 2: 保護ポリシ

3.1 保護対象となる情報と操作

保護ポリシが保護対象とする情報は<INFORMATION>，保護対象とする操作は<OPERATION>で定義される．利用者は個人情報の提供がサービス提供者から要求されると，要求された情報をサービス提供者がどのように処理しようとしているかを利用ポリシとして受け取る．これにより，サービス適用者がどのような情報をどのように処理しようとしているか知ることができる．利用ポリシと<INFORMATION>，<OPERATION>を比較することで複数存在する保護ポリシから適用可能な保護ポリシを絞り込むことができる．

3.2 利用者に対する説明文

保護方法の説明文は<EXPLANATION>で定義される．<EXPLANATION>は保護ポリシがどのように情報を処理するためのものであるかが自然言語で記述される．このため，利用者は<INFORMATION>と<OPERATION>により絞り込まれた保護ポリシの<EXPLANATION>を確認することで，<INFORMATION>の情報を<OPERATION>の処理をどのような方法で実現しようとするものなのか知ることができ，その中から自分が安心できる処理方法を実現する保護ポリシを選択することができる．

3.3 プログラム変換ルール

本モデルではプログラムを変換することで個人情報を保護する．そのため，個人情報（データ）を保護するための形式に変換することが必要となる．データを変換するためのルールは<DATA-CONVERT-RULE>として定義する．また，変換されたデータに対して適用できるようにデータに対する操作を変換することが必要となる．このようなルールを<OPERATION-CONVERT-RULE>として定義する．これにより，保護対象となる情報を変換することができ，保護対象となる情報に対し処理していた操作をデータ変換後のデータに対し処理するように変換する．

また，ネットワークサービスではデータの送受信が行われる．しかし，<DATA-CONVERT-RULE>と<OPERATION-CONVERT-RULE>を適用だけではデータ変換前のデータの送受信が行われてしまい，変換前のデータを保護することができない．そのため，変換前のデータの送受信を阻止する必要がある．また，データ変換後のデータをネットワークサービスで利用するためには，変換後のデータを送受信する必要がある．そこで，データ変換前のデータの送受信を阻止し，変換後のデータの送受信をできるようにするための制御するルールを<COMMUNICATION-RULE>として定義する．<COMMUNICA-

TION-RULE>により，サービス提供者と利用者間のデータフローを制御する．これにより，変換前の情報の送信を阻止し，変換後のデータを送信できるようになる．

4 利用ポリシ

利用ポリシには，サービス提供者による個人情報の処理方法が定義される．プログラムで行われる処理内容を示すために，利用ポリシではどのような個人情報 (データ) がどのような変数に格納され処理されるか，また，そのデータの処理方法を定義する．図 3 に利用ポリシを示す．利用ポリシは<INFORMATION>と<VARIABLE>，<OPERATION>のタグから構成される．

```
<INFORMATION>information<INFORMATION>
<VARIABLE>
   <NAME>name</NAME>
   <TYPE>type</TYPE>
</VARIABLE>
<OPERATION>
   <FORMAT>
      <METHOD>(*<PARAMETER>)
   </FORMAT>
   <METHOD>
      <NAME>name</NAME>
      <RETURN>type</RETURN>
   </METHOD>
   *<PARAMETER>
      <DATA>information</DATA>
      <NAME>name</NAME>
      <TYPE>type</TYPE>
   </PARAMETER>
</OPERATION>
```

図 3: 利用ポリシ

<INFORMATION>はプログラム中で処理する情報の中の１つを定義する．これを読込むことで，この利用ポリシがどの情報を対象として書かれた利用ポリシか知ることができる．<VARIABLE>はプログラム中で使用される変数の情報が定義され，<NAME>と<TYPE>から構成される．<NAME>は変数名が定義され，<TYPE>は変数の型が定義される．<VARIABLE>はプログラム中で<INFORMATION>の情報がどの変数で利用されているかを定義する．保護ポリシには変換対象のデータが記述される．しかし，プログラム中でそのデータがどの変数に格納されているかわからない．そこで，利用ポリシの<VARIABLE>と保護ポリシの変換対象のデータを結びつけることで，データが格納される変数を知ることができる．

<OPERATION>は<FORMAT>と<METHOD>，<PARAMETER>のタグから構成され，<OPERATION>は<INFORMATION>に関係する操作がすべて定義され，<FORMAT>はメソッドがどのように引数を与えているかを示す．<NAME>は名前を示し，<RETURN>は戻り値の型を示す．<PARAMETER>は<DATA>と<NAME>，<TYPE>から構成され，引数の情報が定義される．<DATA>は変数に格納される情報を示し，<NAME>は変数名，<TYPE>は変数の型名を示す．<OPERATION>はプログラム中で<INFORMATION>の情報に対する処理の方法を定義する．<OPERATION>を見ることでプログラムに適用可能な保護ポリシを絞り込むことができる．

5 プログラムの変換

利用者が選択した安心できる処理方法を適用するためには，プログラムを保護ポリシと利用ポリシに従って変換する必要がある．保護ポリシには利用者が守りたい情報とプログラム変換ルールが定義される．また，利用ポリシにはプログラムがどのような情報に対しどのような処理がされるか定義される．プログラム変換モジュールは利用ポリシとプログラム変換ルールを元に個人情報処理プログラムを変換することで保護ポリシを個人情報処理プログラムに反映させる．図 4 にプログラムの変換例を示す．

プログラム変換モジュールは Java プログラムの変換を対象としており，テキスト形式で書かれた保護ポリシと利用ポリシを読込み，プログラムの変換を行う．図 4 の変換前のプログラムは名前を入力して送信するものである．そのため，利用ポリシには，名前が格納され処理される変数が定義される．(名前を対する処理は通信以外に行われていないため，<OPERATTON-CONVERT-RULE>は定義されていない．) また，利用者が選択した保護ポリシには"name"を暗号化して通信することにより守ると定義されている．"name"を暗号化して通信する方法の実現方法は<CONVERT-RULE>に定義されている．<DATA-CONVET-RULE>では<INFORMATION>の"name"を"convert"という関数で変換し"convert-name"に格納し処理するということを示す．<COMMUNICATION-RULE>では"name"の通信を阻止し，"convert-name"を通信することを示す．プログラム変換モジュールの流れを図 5 に示す．

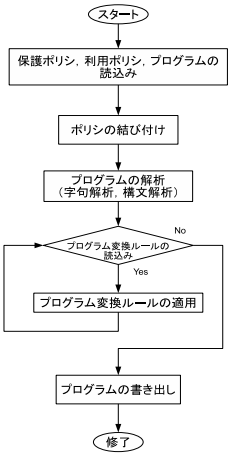

図 5: プログラム変換モジュールの流れ

プログラム変換モジュールは保護ポリシと利用ポリシ，個人情報処理プログラムを読込み，読込まれた保護ポリシと利用ポリシを結びつける．保護ポリシの<INFORMATION>に利用者が守りたい情報が定義され，利用ポリシの<VARIABLE>にデータがどの変数に格納されて処理されるか定義される．そのため，これらを結びつける．これにより，<INFORMATION>に定義された"name"がプログラム中で<VARIABLE>に定義された"data"に格納されて

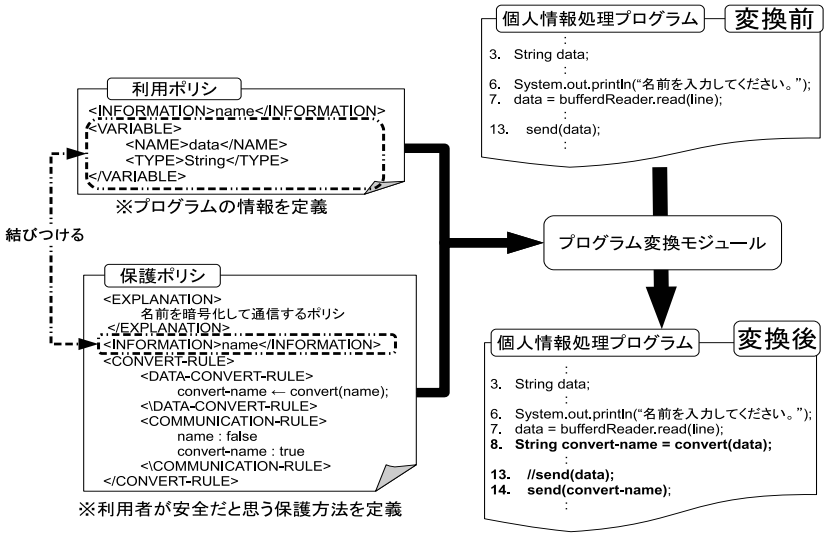

図 4: プログラムの変換例

処理されることがわかる．そのため，<CONVERT-RULE>内の"name"は，プログラム中で変数"data"に格納されていることがわかる．

次に読込んだプログラムを ASTParser[2] を用いて解析する．ASTParser は抽象構文木を生成し，プログラムを構造化する．ASTParser を用いることで，データフローを追跡することが可能となる．構造化されたプログラムを図 6 に示す．

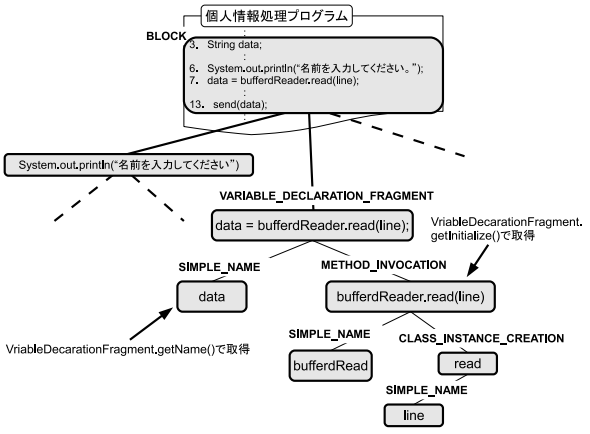

図 6: ASTParser により構造化されたプログラム

各ノードを図 6 のように識別子を付け，この識別子に対応したクラス内のメソッドを呼び出すことで特定の情報を検索することが可能となる．その後，プログラム変換ルールに従ってプログラムを変換する．

図 4 の例ではプログラム変換ルールは<DATA-CONVERT-RULE>と<COMMUNICATION>からなる．<DATA-CONVERT-RULE>はデータを変換するため，"name"が格納される変数"data"をプログラム中から探す必要がある．そのため，プログラムから変数"data"を検索する．検索して特定した行の次の行に<DATA-CONVERT-RULE>に定義された"convert-name = convert(name)"を挿入する．<COMMUNICATION>も同様に，送信箇所を検索し，"name"の送信箇所をコメントアウトし，"convert-name"を送信する行を挿入する．すべてのルールの適用することでプログラムの変換が完了する．最後に変換できたプログラムを java 形式で書き出す．その後に，コンパイルしてプログラムを実行することで，利用者は自身が選択した保護ポリシに従って名前を守ることが可能となる．

6 おわりに

本研究では，プログラム変換モデルを実現するために保護ポリシ，利用ポリシ，プログラム変換ルールを定義し，それらによるプログラムの変換方法を検討した．プログラムの変換では，字句解析や構文解析，プログラム上でデータの遷移を追跡することで，プログラム変換ルールの適用箇所を決定する．プログラム変換ルールの適用を繰り返すことでプログラムの変換を実現する．これにより，利用者は自身が安心できる処理方法で個人情報を処理することができる．今後の課題として，プログラム変換モジュールの実装と評価が挙げられる．

謝辞

本研究は科学研究費補助金（23700098）の支援を受けて行なった．

参考文献

[1] NPO 日本ネットワークセキュリティ協会，2009 年情報セキュリティインシデントに関する調査報告，
http://www.jnsa.org/result/incident/2009.html

[2] Eclipse の ASTParser を試す，
http://www.ibm.com/developerworks/jp/opensource/library/os-ast/