プライバシ情報セキュア流通基盤における プライバシ情報開示制御の実現

西村　祥治†　　宮川　伸也†　　森　拓也†　　佐治　信之†

日本電気株式会社　サービスプラットフォーム研究所†

1. はじめに

サービスの高度化に伴い，より利用者のニーズに合致したサービスやコンテンツを提供するために，携帯電話などの端末を用いて利用者から収集した個人情報や日々の行動情報の活用が注目されている．一方でこれらの情報はプライバシ情報であるため，それらを保護することも重要である．そこで，我々は，プライバシ情報を保護しつつ，その利活用を図るプライバシ情報セキュア流通基盤(Privacy Information Secure eXchange，以下 PISX)の開発，実証を進めている．本稿では，その基盤におけるプライバシ情報開示制御の実現方法について述べる．

2. プライバシ情報セキュア流通基盤

PISX 基盤の目的は，利用者（プライバシ情報提供者）が提供したプライバシ情報の開示範囲をコントロールしつつ，サービス提供者ができる限りプライバシ情報の利活用を図れるようにすることである．前者に関しては，

a　許可されたサービス提供者に，許可された範囲のプライバシ情報だけを開示

b　利用者が開示範囲を判断できるように，誰にどのような情報が提供したかを追跡

を実現し，後者に関しては，

c　関連するサービスへプライバシ情報のセキュアな利活用の実現

d　セキュアな利活用を容易に実現できるプログラミングモデルを提供

を図った．

PISX 基盤の構成は図 1の通りである．利用者から集められたプライバシ情報は，各利用者の開示ポリシに紐付けられて行動情報履歴 DB に蓄積される．サービス提供者は，利用者にサービスを提供するにあたり，行動情報履歴 DB よりプライバシ情報を取得する．このとき，プライバシ情報は秘匿（暗号化）された状態で流通される．そして，利用の際，開示ポリシ(a)にしたがってプライバシ情報が開示されるとともに，操作履歴が記録される(b)．サービス事業者はプライバシ情報を加工し，関連するサービスに提供することも可能である．このとき，強制的に該当するプライバシ情報を秘匿することで，セキュアな流通を実現する(c)．サービス事業者は，プライバシ情報を利用するにあたり，開示秘匿制御の機能を組み込む必要があるが，暗黙的に自動開示・秘匿する機能を提供することでサービス開発の負荷を軽減する(d)．

利用者によるプライバシ情報の開示範囲のコントロールは，[1]で解説し，本稿では，サービス提供者のサービスにおけるプライバシ情報の開示・秘匿制御について説明する．

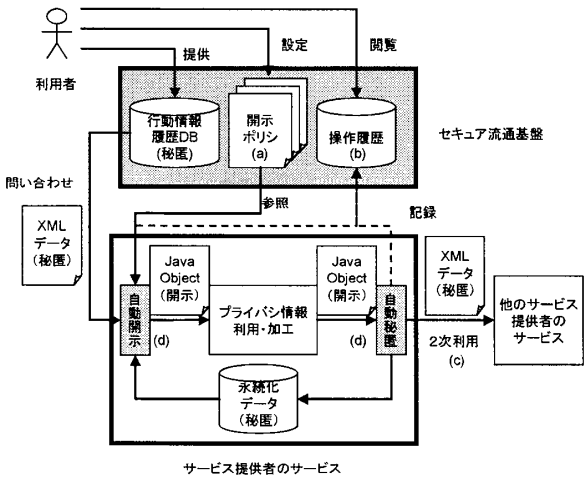

図 1 PISX 基盤構成図

3. プライバシ情報開示制御の実現

サービス提供者のサービス（以下，サービス）におけるプライバシ情報の開示・秘匿制御の実現方法について述べる．実現にあたっては，プライバシ情報が開示される区間を短くし，漏洩のリスクを減らすこと，既存のサービスに対する影響（コード修正など）を限定し，容易に適用できることを目指した．なお，本基盤およびサービスは実装言語として Java を用いた．

3.1. プライバシ情報の表現

プライバシ情報は，以下の特徴を持つ．

a　データの表現形式が開示形式（生データ）と秘匿形式（暗号化データ）の２つある

b　プライバシ情報の種類，所有者などのメタ

An Implementation of Privacy Information Disclosure Controls on Privacy Information Secure Exchange Platform

† Shoji Nishimura, Shinya miyakawa, Takuya Mori and Nobuyuki Saji (Service Platforms Research Labs, NEC Corporation)

データを持つ

安全性を高めるためには，プライバシ情報はプログラムで操作するときだけ開示形式で，それ以外は秘匿形式であることが望ましい．そこでプライバシ情報の型は両形式が共存できるString 型に制限し，必要なときだけ開示形式として表現できるようにした．

また，プログラム中では，既存のサービスへの影響を最小限にして，プライバシ情報に関するメタデータを保持する必要がある．そこで，プライバシ情報が入るフィールドにプライバシ情報が入ることを示すアノテーションを付与し，利用者 ID など管理用のデータを格納するフィールドを持ったクラスを継承することで、メタデータを保持した．これらによりサービスのロジックに影響を与えずメタデータを保持することができる．

3.2. 自動開示・秘匿

前節で述べたようにプライバシ情報は開示形式と秘匿形式の２つあり，プログラム中で利用するには，適切なタイミングで形式を変換する必要がある．開示区間の最小化の観点からは，サービス実装者により明示的に開示・秘匿を指示させたほうがよいが，

a　サービスのロジックの修正が必要でサービス実装者の負荷が大きい

b　意図のあるなしにかかわらず実装ミスによる漏洩の可能性が高い

などの問題がある．そこで，本方式では，XML 変換ライブラリ(XStream[2])，Object Relational Mapping ライブラリ(Hibernate[3])に開示・秘匿処理を組み込む方式を採用した．なぜなら，

a　ネットワーク(XML 形式の電文)と DB などサービスの外部との接点である

b　ライブラリ交換で PISX の適用が可能であるからである．

XStream や Hibernate は，JavaBeans 形式の Java オブジェクトと相互にデータ変換するように実現されている．そこで，変換時に前節で述べたアノテーションや管理用フィールドを手がかりにして，プライバシ情報の発見および開示形式・秘匿形式の変換を実現する．これによりサービス実装者はいつ開示・秘匿されることを意識することなく，単にライブラリを交換することで PISX の適用が可能となる．

図 1で示すようにサービスへは必ず秘匿形式でプライバシ情報が渡され，PISX 化されたライブラリを用いなければ開示することはできない．この結果，サービスがプライバシ情報を利用するには PISX 化されたライブラリを使用することが強制されることになる．

また，開示・秘匿処理時に PISX 基盤に操作履歴を送信する．これにより，利用者はどのサービス事業者がどのプライバシ情報を利用したかを追跡することが可能となる．

4. 評価

情報大航海プロジェクトのモデルサービスにおいて，あるサービス呼び出しにおける開示・秘匿のオーバヘッドの計測を行った．計測対象のシーケンスは，DB からプライバシ情報を取得（および開示）し，それをそのまま電文メッセージに変換して送信（および秘匿）するものである．その３８回分の結果を，表 1にまとめる．評価対象としたシーケンスでは，DB の内容を電文に変換する単純なものであったため，XML へ変換時の秘匿処理が大きく見える結果となった．しかし，データの統計解析などプライバシ情報の加工を必要とする場合は，開示・秘匿処理に占める割合はこれより低下するため，ほぼオーバヘッドの最大値と考えられ，通常は２～３割程度になると見込まれる．

表 1 開示・秘匿処理のオーバヘッド

	累積時間 (sec，38 回分)	割合 (%)
秘匿(XML へ変換時)	248.131	39.4
開示(DB から読出時)	44.243	7.0
トレースログ	7.948	1.3
ロジック部	329.512	52.3
	629.834	100.0

5. まとめ

PISX におけるプライバシ情報開示・秘匿制御の実現方式について述べ，比較的容易にサービスへ適用できることを示した．また，適用した際のオーバヘッドの評価を行い，その最大で 40%強あることを示した．今後は，オーバヘッドの低減やより開示区間の短縮や容易でセキュアな PISX の導入方式をすすめていきたい．

本研究は，経済産業省「情報大航海プロジェクト」のモデルサービスとして，株式会社 NTT ドコモを中心とした「マイ・ライフ・アシストサービス」実証実験の一環として実施した．

参考文献

[1] 宮川伸也　他，“提供者が主導となりプライバシ情報の開示制御が行えるプライバシ情報セキュア流通基盤の実現”，第７１回情報処理学会全国大会 5E-2(発表予定)

[2] XStream http://xstream.codehaus.org/

[3] Hibernate http://www.hibernate.org/