タイムロックメッセージを対象としたWebサービス連携方式の実験と評価

安田　恭行†　　中所　武司†

明治大学 理工学部 情報科学科

1. はじめに

競争入札の公告や開札，選挙の電子投票の開票，試験問題やその結果の公表など，電子文書の公平な公開を必要とする場合に，あらかじめ暗号化された電子文書が将来の指定時刻まで復号されないことを保証するシステムが有効である．そこで，このようなタイムロックメッセージシステムを対象に，複数の単一サービスを組み合わせて複合サービスを実現する方式を実装した．

具体的には，暗号化サービスと復号化サービスと時刻発信サービスを実装するとともに，これらを統合してタイムロックメッセージサービスを実現した．実装技術としては，Java，Apache Axis2，GPG，SOAP，WSDLなどを用いた．

2. Webサービス連携技術の位置づけ

インターネットの普及とともに，Web アプリケーションが増大し，ASP や Web サービスに加えて，SOA，SaaS が注目されるなど，ソフトウェアのサービス化が促進されている．

我々は，変化の激しい時代には，エンドユーザ主導のアプリケーション開発とその保守が重要になるという観点から Web サービス連携の研究を行ってきた．特に，小さな部門や個人の業務を対象とする中小規模の Web アプリケーションに関して，低コストで短期間に開発するとともに，頻繁な機能変更を伴う保守にも対応するために，その分野の業務の専門家主導で開発・保守ができるような技法を研究してきた[1]．

3. 例題システムの概要

タイムロックメッセージサービスのシステム構成を図1に示す．メッセージ送信者は，指定時刻に見てほしい文書を暗号化するとともに，この暗号化の鍵をさらに暗号化し，これらの暗号化された文書と鍵をメッセージ受信者に渡す．メッセージ受信者は，指定時刻に達すると，鍵を復号し，その鍵で文書を復号する．

この鍵を復号する際に時刻認証を行い，指定時刻まで鍵を復号できなくすることで，指定時刻まで文書を復号できない仕組みを実現する．

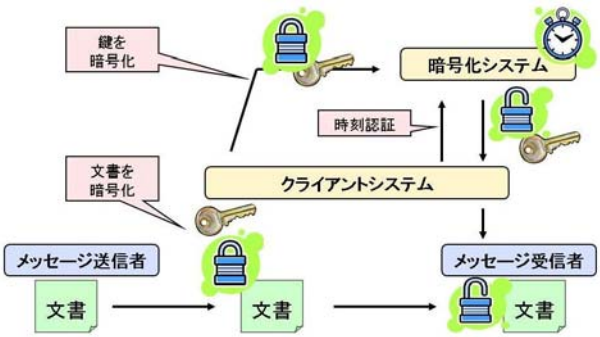

図1　システム構成

4. 個別のサービスの実現

4.1 実装方式

この例題を Web サービス連携で実現するために，各サービス間のやりとりには SOAP を用いた．SOAP はネットワーク上のオブジェクトへアクセスするためのプロトコルであり，SOAP による通信では，エンベロープと呼ばれる XML 文書に情報を格納し HTTP などのプロトコルで交換する．サービスを利用するクライアントとサービスを提供するサーバの双方が SOAP の生成・解釈エンジンを持つことで異なる環境間でのオブジェクト呼び出しを可能にしている．

本研究では，Apache Axis2 を用いて個別のサービスを構築した．Axis は SOAP の生成・解釈エンジンを構築するためのフレームワークである．Java で作成されており，サーブレットとして Tomcat 上で動作する．そこに Web サービスとして提供したいクラスを配置することで外部からそのサービスを呼び出せるようになる．

4.2 暗号化サービスと復号化サービス

入力されたファイルを暗号化して出力するサービスでは，暗号化ソフトウェアとして GPG[2] を用いた．java.lang.Runtime クラスの exec メソッドを外部コマンドとして呼び出している．

Web service integration for time lock message

† Yasuyuki Yasuda and Takeshi Chusho,
Department of Computer Science, Meiji University

さらに，サービス間のファイルのやりとりのために，SOAP にファイルを添付する SOAP Messages With Attachments[3]を用いてインターフェイスを実装した．

暗号化されたファイルを復号するサービスも，暗号化サービスと同じ実装技術を用いた．

4.3 時刻発信サービス

現在時刻を発信するサービスでは，現在時刻を 1970 年 1 月 1 日 0 時 0 分 0 秒からの経過時間をミリ秒に直した整数値を返す．NTP (Network Time Protocol)を使用してサーバの時計を日本の標準時刻に合わせ，Date クラスを使い，その時刻を取得している．

このサービスで入手する現在時刻と暗号化された文書の指定時刻を比較して時刻認証を行う．

5. 複合サービスの実現

5.1 実装方式

タイムロックメッセージサービスは，図 2 に示すように 3 種類の単一サービスを連携させて実現した．Apache Axis2 に含まれるツール WSDL2Java を使い，WSDL をもとにサービスを利用するためのクラス(スタブ)を自動生成し，各サービスの呼び出しを行った．

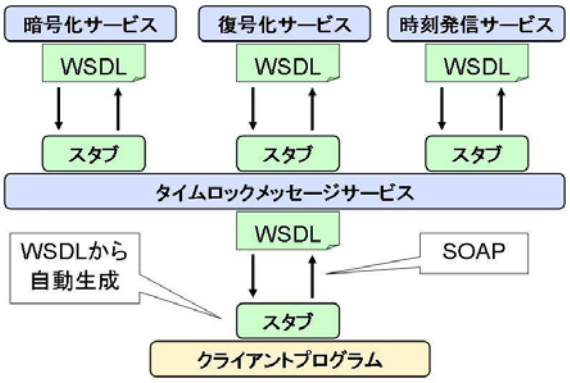

図 2　サービスの連携

5.2 文書の暗号化と復号化

実際のサービスの流れを図 3 に示す．メッセージ送信者から指定時刻と文書を SOAP のパラメータとして受け取り，その文書を暗号化する(1)．この時使用した鍵を指定時刻とともに暗号化サービスを用いて暗号化して(2, 3)，SOAP に添付し出力する(4)．送信者はこのデータを受信者へ直接渡す(5)．

受信者は指定時刻になるとこのデータを入力する(6)．そのデータを復号化サービスで復号して暗号化の鍵と指定時刻を取り出す(7, 8)．同時に時刻発信サービスから現在時刻を取得し，両者を比較する(9, 10)．指定時刻に達していれば文書を復号し出力する(11)．

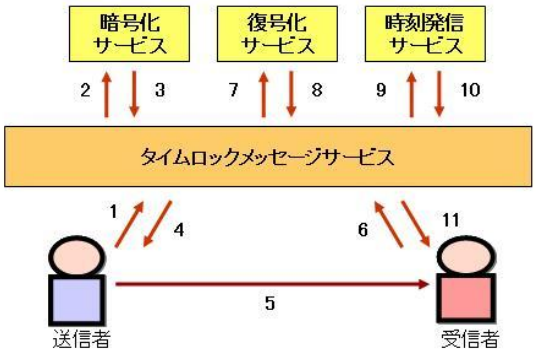

図 3　処理の流れ

6. Web サービス連携に関する考察

一般に Web サービスは独立して開発され，プラットフォームや開発言語，開発環境等に依存しないため，個々のサービスに適した条件で開発を行える．これらの個別サービスを利用することにより，効率よく複合サービスを開発できることを確認した．関連技術の整備や開発環境の充実で SOAP を意識せずに開発を行えるようになっていきているので，Web サービスの構築や利用は容易になっているが，連携部分でのプログラミングの知識は必要である．

そこで，業務の専門家主導の開発・保守実現のために，「1 サービス＝1 フォーム」という観点で，入力フォームから出力フォームへの変換定義により複合サービスを構築する方式[1]を現在研究中である．

7. おわりに

実行性能は，この 2 頁の論文をデータにした場合の暗号化・復号化とも 1 秒以内，5MB のテキストでも 2 秒程度で，実用上の問題はない．

参考文献

[1] 中所武司：業務の知識を有するエンドユーザ主導のアプリケーション開発技法，電子情報通信学会 知能ソフトウェア工学研究会 KBSE2007-30, 19-24(Nov. 2007).

[2] GnuPG: The GNU Privacy Guard http://www.gnupg.org/

[3] W3C: SOAP Messages With Attachments http://www.w3.org/TR/SOAP-attachments