

緊急災害時に有用な 家族間の個人情報共有システムの提案と実装

長谷川 友香[†] 小口 正人[†]

[†]お茶の水女子大学 理学部 情報科学科

1. はじめに

東日本大震災のような大きな災害に遭った場合、家族の安否を第一に確認したいと思うであろう。ところが、このような災害時には電話回線が混雑し、家族と直接連絡がとれないことが多い。

そのような状況において、スケジュール情報や移動履歴、メールなどの個人情報を得ることができれば、家族が災害時にどこにいたかを特定する大きな手がかりとなり有用である。しかし、家族とはいえ平常時から個人情報が閲覧できてしまうことには抵抗があると思われる。さらに、大規模な災害ではなくても、子供と連絡がつかない場合にはスケジュールだけ閲覧できるようにするなど、緊急の度合いに合わせて閲覧できる情報を制御したいということが考えられる。

そこで、本研究では個人情報を共有するシステムを構築し、緊急時のみ閲覧を許可する認証方法を実装する。その認証方法としては階層型相互認証を提案する。階層型であるため、情報の閲覧を許可するかしないかの切り替えではなく、どのレベルの情報まで許可するかを制御可能となる。

2. 階層型相互認証

各ユーザが他のユーザ各々に対して認証レベルを保持する。スケジュールや移動履歴などの情報に対して閲覧を許可する認証レベルを割り当てる。レベルが高いほど閲覧されにくい情報となる。他のユーザと自分が同じ認証レベルにいる時のみ、そのレベルの情報閲覧可能である。

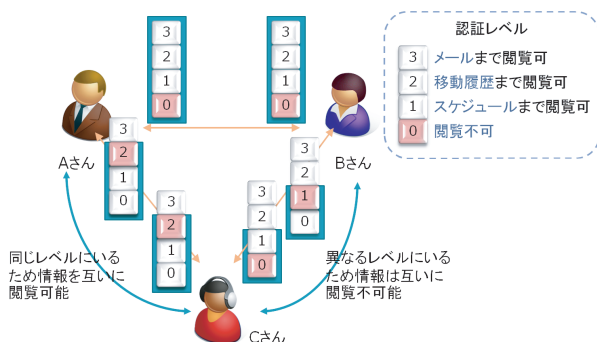


図 1: 階層型相互認証の概要図

緊急時に他のユーザの情報を閲覧する必要性が生じた場合、他のユーザの自分に対する認証レベルを変更することができる。ただしこのとき、レベルを変更したユーザ

の相手に対するレベルも同じところまで上がって相手からも同じレベルの情報が閲覧される状態になり、レベルを変更されたユーザに対して通知される。これにより、通常時にむやみに他のユーザの情報を閲覧することへの抑制になると考えられる。また、二人の間で上げられる認証レベルの最大を設定し、緊急時でもそのレベルより上には上げられない仕様とする。この階層型相互認証の概要図を図1に示す。

3. 開発環境

本研究では Google App Engine と Android 端末を使用してシステムを構築する。システム構成図を図2に示す。

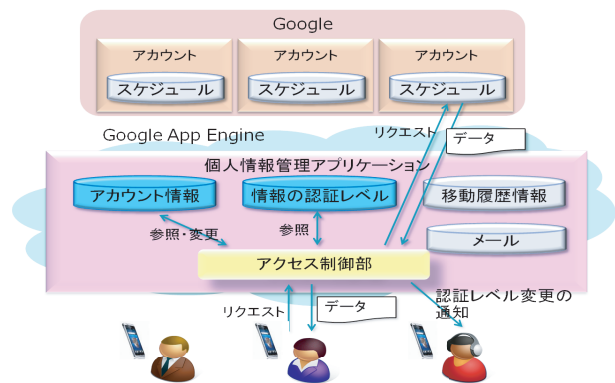


図 2: システム構成

3.1 Google App Engine

Google App Engine とは Google 社が提供するクラウドサービスで、PaaS 形式のアプリケーションプラットフォームである。Python や Java, Go 言語を使用していることが可能であり、本研究では Java を用いて開発した。Java で開発する場合は Web サイトの構築に Java Servlet と呼ばれる技術を用いる。データの保存にはキーバリューストア型のデータベース管理システム「Big Table」をベースとした Datastore と呼ばれるサービスを用いる。Google App Engine にユーザ情報や認証レベル、移動履歴情報などを保存し、階層型認証の制御を行う。ユーザは Google App Engine 上に作成した Web サイトから情報の閲覧ができる。

3.2 Android

Android は、Google 社を中心とするモバイル機器の共通プラットフォームを推進する組織「OHA (Open Handset Alliance)」によって発表された携帯情報端末向けのプラットフォームである。本システムではこの Android を搭載したスマートフォンを用いて移動履歴の取得と情報の閲覧を行う。

4. 個人情報へのアクセス

本研究で扱う個人情報はスケジュール、移動履歴、メールの3つである。通常の利用であればこれらの情報にアクセスするのは本人だけであるが、情報共有のためにはシステムからアクセスする手法が必要になる。それぞれの情報へのアクセスの実装方法を以下に示す。システムから各ユーザの個人情報へのアクセスを可能にした上で、認証を行ってそれらの情報へのアクセスを制御する。家族内に特権ユーザは存在せず、システムが認めたレベルまでの情報を互いに閲覧可能となる。

4.1 スケジュール

スケジュール情報としては、Google Data API を用いて Google カレンダーの情報を取得する。Google Data API とは REST 形式でカレンダーやマップなどの Google のリソースを操作できる API である。本システムでは、この API を用いるに当たり OAuth 認証を行う。OAuth 認証とは、あらかじめ信頼関係を構築したサービス間でユーザの同意のもとにセキュアにユーザの権限を受け渡しする「認可情報の委譲」のための仕様である。本システムの初回利用時にユーザがシステムに対して Google カレンダーへのアクセス権限を委譲し、システムはそれ以降その権限を使って個人のカレンダー情報を取得することができる。

4.2 移動履歴

移動履歴は Android 端末で GPS を使って取得する。具体的には、Android 端末で動くアプリケーションを作成し、一定時間ごとに現在地の緯度と経度を取得してサーバに送信させる。サーバはその情報を蓄積し、閲覧のリクエストがあった場合には Google Map 上に位置を線でつないで表示する。

4.3 メール

メールは転送設定を行い、クラウド上のアプリケーションで受信して蓄積する。本システムでは Gmail からの転送を想定して実装している。

5. 階層型相互認証機構の実装

本章では階層型相互認証機構の実装方法について説明する。クラウドに保存するデータクラスとして、組となる二人のユーザ ID、二人の現在の認証レベル、認証レベルの上限値を保存するクラスを作成した。Google App Engine でこのようなクラスから作られるオブジェクトが Datastore に保存されるとエンティティと呼ばれるようになる。このエンティティの数は、システムのユーザの総数を n とすると nC_2 となる。新規のユーザが登録されるとともにエンティティの数は増えるので、新規ユーザと既登録の各ユーザを組とするエンティティを作成する。

他のユーザの一覧ページの表示をリクエストされると、システムは操作中のユーザ（以下、操作ユーザ）を含むエンティティを探し、そのユーザと対になっているユーザ（以下、相手ユーザ）の一覧を作成する。その一覧には相手ユーザの ID、相手ユーザと操作ユーザの現在の認証レベル、認証レベルの上限値を表示し、閲覧可能と



図 3: 登録済みユーザ一覧画面

なっている情報名をハイパーリンクとして表示する。このリンクがクリックされると相手ユーザの情報を閲覧するページに遷移する。

相手ユーザの認証レベルを上げるボタンと、相手に対する操作ユーザの認証レベルを設定するリストボックスも表示する。相手ユーザの認証レベルを上げるボタンが押されると、該当するエンティティの相手ユーザの認証レベルと操作ユーザの認証レベルを書き換え、相手ユーザに対して認証レベル通知のメールを送る。このとき、相手ユーザの認証レベルは 1 段階ずつしか上げられず、下げることはできない仕様としている。これは、簡単に相手のレベルを変更できないようにすることで、平常時にレベル変更を行うことの抑制につながると考えるためである。操作ユーザの認証レベルは、リストボックスから選んで自由に変更できるようにする。この変更がなされると、該当するエンティティの操作ユーザの認証レベルのみ書き換える。これら二通りの認証レベルの変更に共通して、認証レベルの上限値より上には上げられないようにしている。

6. まとめと今後の課題

緊急時に家族間で個人情報を共有するためのシステムを Google App Engine と Android 端末を用いて構築した。プライバシーの問題を考慮し、緊急時のみ情報の閲覧が可能にするための階層型相互認証を実装した。

現段階では相手ユーザの認証レベルの変更が即座に反映されるようになっているので、一定時間相手ユーザから応答がない場合のみレベル変更を行うなどより良い認証方法を検討し、実装していきたい。

謝辞

本研究を進めるにあたり、お茶の水女子大学の小林一郎教授、とめ研究所の宇田川浩氏、日立製作所の谷本幸一氏に大変有用な助言をいただきました。

参考文献

- [1] クラウド活用のための Android 業務アプリ開発入門 出村成和, 日経 BP 社, 2011
- [2] すっきりわかる Google App Engine for Java クラウドプログラミング 中田秀基, ソフトバンク クリエイティブ株式会社, 2010