# 適応型認証へのWiFi 応答時間の応用とその改良手法

原 大司[1,a)]　金岡 晃[2]　櫻井 幸一[1]

**概要**：電子認証における研究において近年注目を集めている分野の 1 つに適応型認証がある. 本研究では，位置情報に WiFi 応答時間を用いることによる認証の高精度化と高利便性の実現を目的とした．また，WiFi 応答時間を用いた認証システムを補助する要素として，予定管理ツールによる時間情報，ユーザーのブラウザ設定言語による言語情報を用いた改良について評価を行った．

# Application of WiFi Round Trip Time to Adaptive Authentication and its Improvement Method

## 1. はじめに

インターネットは多くの場面で多くの人々に利用され，様々な情報が電子化されたデータとして取得，発信されている．その中で情報の価値，重要性が見出されてきた．しかし，そのインターネットを通して，個人情報などの重要な情報が意図しない形で，あるいは他者からの攻撃によって知らない間に拡散してしまうかもしれない [1]．よって，情報をやり取りする上での安全性を確保するために，サイバーセキュリティ技術が重要視されてきており，ユーザーがサービスを利用する際に必要になる認証技術に関する研究は，盛んに行われている．認証の安全性を強化するための方式として，単一要素認証，二段階認証，二要素認証，多段階認証，適応型認証などが存在する．

認証に用いる要素として，三大要素が共通認識として存在する．それは知る要素（What You Know），持つ要素（What You Have），備える要素（What You Are）である．認証システムはこれらの要素を用いることによって認証を行う．従来の認証システムは，これらの要素のうち 1 つの要素のみを用いた単一要素認証が主だったが，単一要素認証の脆弱性を突く攻撃が行われた．そこで，現在では二段階認証を採用することが多くなった．

多段階認証は，従来の単一要素認証の認証の段階数を増やすことにより，より安全性を高めた認証システムである．認証の段数を増やすにつれ安全性が高くなるが，認証の段階数を増やすことに伴う利便性の低さが課題となっている．利便性が低下した場合，ユーザー側はアクセスに手間を感じてしまう．その結果，ユーザーの安全性意識が低下し，脆弱なパスワード設定を行うことで，少しでも認証に対する負担を軽くしようとする．攻撃者はこのような脆弱性を標的にし，不正アクセスを試みる．このような状況から，認証システムの利便性が重要視されている．

利便性の課題を解決する認証方式として，現在注目を集めている研究の 1 つに適応型認証（Adaptive Authentication）がある．適応型認証とは，様々な認証状況に応じて最良の認証方法をシステムが選択する認証システムである．適応型認証に関する研究は 2003 年に登場し，利用環境などユーザーの文脈に応じて認証を適応的に行うことで，高い利便性を持つ認証方式となっている．

適応型認証における既存研究では，ユーザーの背景情報として時間情報，位置情報が用いられていた．既存研究の適応型認証で用いられる位置情報は，GPS 機能により取得される情報が使用されている．しかし，GPS は最大で 100 メートル以上の位置誤差が発生する点で問題となる．適応型認証は，一般的に用いられている GPS による位置情報を用いるのではなく，より有効な背景情報を用いることで，認証の高精度化，高利便性が実現できる点で課題がある．

本研究では，WiFi 応答時間 (RTT : Round Trip Time) による位置情報を利用することで，上記の課題を解決する．

[1] 九州大学
[2] 東邦大学
[a)] daiji.hara@inf.kyushu-u.ac.jp

認証の利用環境として，大学において研究室に所属する学生における研究室内リソースアクセスを設定した．WiFi RTT を補助する背景情報として，個人あるいは研究室のスケジュールを利用した時間情報，ユーザーが使用する言語情報を用いる．本研究では位置情報を WiFi RTT を用いて，時間情報を予定管理ツールを用いて，言語情報をユーザーエージェントから取得している．WiFi RTT は WiFi アクセスポイントを 3 個以上設置することにより，1〜 2m の誤差で位置情報の取得が可能であり，GPS の大きな位置誤差による課題を解決する．

WiFi RTT による位置推定を行い，その誤差は最小の地点で 365.649764901748mm，最大の地点で 4532.422314593179mm だった．誤差が最小となる箇所では，その情報のみで認証を行うことが可能な精度であったが，誤差が最大となる箇所では，その情報のみで認証を行うには十分な精度ではなかった．しかし，WiFi RTT による位置情報だけでなく，予定管理ツールによる時間情報，ブラウザ設定言語による言語情報などの他の背景情報と組み合わせた．WiFi RTT を取得する際に追加した機器は，Google Nest WiFi ルーターが 1 機，Google Nest WiFi 拡張ポイントが 3 機，Google Pixel 3a が 1 機である．これらは市販されており，簡単に入手可能である．また，WiFi RTT を取得するための API は Android Developers で公開されており，Android Studio でアプリケーションが作成可能である．予定管理ツールによる時間情報は，Google カレンダーの API を利用することで取得可能である．ブラウザ設定言語による言語情報は，HTTP ヘッダ内の html を抽出することにより取得可能である．これらのことから，WiFi RTT による位置情報，予定管理ツールによる時間情報，ブラウザ設定言語による言語情報は実現可能である．また，これらの背景情報は，ユーザーが認証を行う際に入力する必要が無い．このことから、システムの利便性は高いと言える．

## 2. 関連研究

### 2.1 A Survey on Adaptive Authentication

A Survey on Adaptive Authentication[3] では，Arias-Cabarcos らによって適応型認証に関する研究の現状について調査が行われている．現代では，パスワード中心の認証システムが主流であり，適応型認証は，その環境を変える可能性を孕んでいる．しかし，適応型認証は実用段階に至っていない．そこで Arias-cabarcos らは，適応認型証設計を改善する方法を見出す為に，既存研究を調査，提供している．本研究では，この論文を読むことで適応型認証の現状を把握するとともに，取り上げられている適応型認証研究から，関連論文として 4 つ取り上げている．

### 2.2 UFSA (User-Friendly and Secure Authentication)

UFSA[4] は明示認証子 (explicit authenticators) と暗黙認証子を用いた適応型認証を提案している．明示認証子とは，ユーザー自身が意識して用いる認証子 (例: パスワード，SMS コードなど) であり，暗黙認証子とは，ユーザーが意識をしないで用いられる認証子 (IP アドレス，時間情報など) のことである．ユーザーが認証を行う際に，システムはユーザーの暗黙認証子を確認する．その暗黙認証子を基に，明示認証子にかかる重みスコアを算出する（ここで，暗黙認証子による認証が失敗した場合，重みの影響により明示認証を行う際の困難は大きくなる）．ユーザーが入力する明示認証子にこの重みがかかり，そのスコアが閾値を超える場合にユーザーはサービスを利用できる．閾値を超えない場合，そのスコアギャップを埋める必要があり，追加の認証が行われる．このスコアギャップを埋める最適な認証子の選択手法を UFSA は提案している．このアルゴリズムにより，UFSA は必要最低限の困難になるような認証子の選択を可能にしている．性能を測定するために，Naive 法と Greedy 法との比較を行う．Naive 法は認証子の選択手法として，最も簡単に使用できる認証子の選択を行う．Greedy 法は困難が高い認証子を優先的に選択す る．この比較を行うに当たって，各明示認証子に対して，ユーザーにはあらかじめそれぞれの主観で困難度を設定させておき，管理者はサービスを利用するための閾値 (Operation Sensitivity) を設定しておく必要がある．暗黙認証子による重みと困難度により最終的なスコアが計算され，その値が閾値を超えるとユーザーはサービスを利用できる．UFSA は Greedy 法と比較して最大 18%，Naive 法と比較して最大 20%ユーザーが感じる困難が低いことが結果として得られている．

### 2.3 ASSO (Adaptive Single Sign On)

ASSO[6] はシグルサインオンシステムを適応型にした研究である．この研究の目的は，認証に必要な安全性を保ちつつ，利便性を向上することである．この研究では，168 人からデータを収集している．168 人の参加者がスマートフォンを所持し，一年間に渡って位置情報と時間情報のデータを蓄積した．端末に保存されたデータは特定のアクセスポイントを検知した際にデータベースにアップロードされる． 最初の一ヶ月に収集したデータを訓練データ，残りをテストデータとし，SVM によって分類モデルを作成した．これにより，再現率 87.7%，適合率 88.4%という結果が得られた．既存研究では用いられていない位置情報を認証子に用いている点，実際のモバイルデータを用いた実験を行っていることがこの研究の貢献である．

### 2.4 Smart-Auth

Smart-Auth[5] では，ForgeRock 社が提供する OpenAM 上に実装する適応型認証のフレームワーク SmartAuth を提案している．SmartAuth は，認証に用いる認証子の選択を Hoeffding 木を用いて行っている．Hoeffding 木はデータストリームを対象とした決定木学習アルゴリズムである．これにより最適な認証子を選択できような木を逐次更新していくことができるため，適応性を獲得している．SmartAuth では，データを類似性保存ハッシュ関数に通してフィンガープリントに変換したものを，認証子として利用している．データはユーザー側とサーバー側から集められる．ユーザー側から集められるデータには言語，色深度，画像解像度などがあり，ユーザーはこれらのデータのうち認証システムに提供するものを選択することができる．サーバー側から集められるデータには IP アドレス，アクセス時刻，位置情報などが存在する．作成したフィンガープリントに対しタイムスタンプなどの情報を加えることで，その認証子を用いて認証を行なうか否かの判断が行われる．この手法により，99%の精度で良悪性の分類を可能とした．これを 6 人に対してユーザー実験を行ったところ，被験者達は概念レベルで SmartAuth の価値を理解した．しかし，被験者達は技術的な背景を持ち合わせており一般人とは言えない点，ユーザー実験には人数が不十分である点で課題が残った．

### 2.5 Reinforced-Auth

Reinforced-Auth[7] は，疑わしいログインを識別するフレームワークの開発を行っており，大規模データセットを効率的に評価できるプロトタイプを実装している．ユーザーがログインを試行したとき，そのユーザーがのログイン履歴，用いている認証子を確認する．それらの情報を基にスコアを計算し，それによりログインを 3 種類に分類する．ログインが良性か悪性か判断がつかない場合，追加の情報を要求する．この実験のデータは LinkedIn の一年間のログデータを使用している．このログデータを良性ログイン，不正アクセス，ボットによるログインの 3 種類にラベル付けして利用している．このデータを用いて，以下の式でスコアを計算しログインの良悪性を分類している．この実験では，偽陽性率を 10%固定で真陽性率を計算することで ROC 曲線を描き，その AUC により性能を測定している．これにより，ボットネットに対しては 10%の偽陽性率で 95%の検知が可能であり，アカウント侵害に対しては 10%の偽陽性率で 77%検知が可能という結果が得られた．この実験で使用した認証システムと同様の方式が既に大規模ウェブサイトで利用されているが，この研究はこの方式を最初に分析し，ベンチマークを提供したという点で貢献している．

### 2.6 課題

上記の既存研究は，背景情報として GPS による位置情報，時間情報，IP アドレス，ユーザーエージェントを利用している．これらの研究は共通する背景情報を用いているので，認証状況を特定することにより，認証の高精度化，高利便性の可能性を孕んでいる．また，位置情報に GPS を用いていることによる大きな位置誤差 [8] が課題として挙げられる．

## 3. 貢献

適応型認証は，ユーザーの背景情報に応じて認証を適応させるシステムである．既存研究では，背景情報に位置情報を用いており，それは GPS により取得されている．GPS による位置情報は，大きな位置誤差が発生するという点で課題がある．既存研究のシステム構成図の概形は，図 1 で表すことができる．本研究では，図 1 中の暗黙認証子として用いる背景情報の選択に着目した．本研究では，位置情報を WiFi RTT により取得することで，既存研究の課題を解決する．また，WiFi RTT による位置情報を補助する要素として，予定管理ツールによる時間情報，ユーザーのブラウザ設定言語による言語情報を利用することで，認証の高精度化，高利便性を実現する．

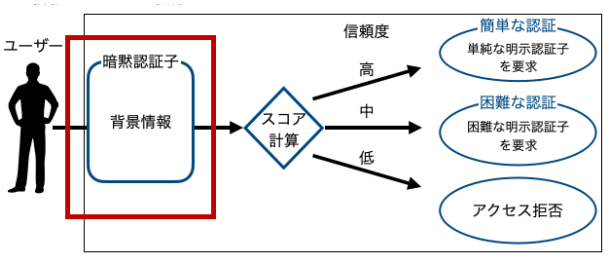


**図 1** 既存研究のシステム構成図の概形

## 4. 認証環境により発生する背景情報の利用

適応型認証における既存研究では，位置情報を GPS により取得しており，大きな位置誤差が発生する点で課題がある．そこで本研究では位置情報を WiFi RTT によって収集することで，既存研究の課題を解決する．また，WiFi RTT による位置情報を他の背景情報と組み合わせることで，システムのさらなる高精度化を図る．そこで本研究では，認証環境により発生する背景情報を WiFi RTT による位置情報を補助する要素として用いた．本研究では，認証環境により発生する三つの特性を利用した背景情報を用いている．

### 4.1 認証環境

本研究では認証環境として，大学において研究室に所属する学生における，研究室内リソースアクセスを設定した．

研究室に所属する学生は，研究室内における位置的特性，研究室内に滞在する時間帯による時間的特性，所属する学生による言語的特性の三つの特性を持つと考えられる．

#### 4.1.1 位置的特性

研究室に所属する学生は，位置的特性を持つと考えられる．研究室内では，自分の研究を行うデスク，ゼミ等の作業スペースなどある程度の行動範囲が定まっている．このような研究室内での位置的特性を位置情報として用いることで，ユーザーの同定を行うことが可能であると考えられる．研究室内等の広さの範囲では，GPS による位置情報は，その大きな位置誤差により利用することができない [8]．本研究では位置情報を WiFi RTT[9] を用いることにより取得する．WiFi RTT による位置情報を用いることで，位置誤差 1~2 メートルの範囲で位置情報を取得することができると言われている [9]．また，WiFi ルーターなどの WiFi アクセスポイントとスマートフォンなどの WiFi 受信端末があればこの位置情報の取得が可能であり，特殊なセンサー等の機材を用意する必要がない．

#### 4.1.2 時間的特性

研究室に所属する学生は，時間的特性を持つと考えられる．学生はその活動として，研究室内でのゼミ，自身が受講している科目の講義や TA 業務等が存在し，これらの時間的制限により，研究室に滞在する時間帯についてある程度の傾向を持つ．このような時間的特性を時間情報として用いることで，研究室内リソースにアクセスするユーザーの本人性の信頼度合いを判断することができる．この時間情報は予定管理ツールを用いることで取得可能であり，特殊なセンサー等の機材を用意する必要がない．

#### 4.1.3 言語的特性

大学における研究室では，国内からの学生だけでなく，海外からの留学生や研究生が配属されるため，研究室に所属する学生は多国籍である．そのため，ブラウザに設定する言語にばらつきが生じるので，この点において言語的特性を持つ．この言語特性を言語情報として用いることで，研究室内リソースにアクセスするユーザーの本人性の信頼度合いを判断することができる．この言語情報はブラウザに設定されている言語情報を，HTTP ヘッダに存在する html により取得可能であり，特殊なセンサー等の機材を用意する必要がない．

## 5. 実験

### 5.1 実験内容

適応型認証はユーザーの背景情報に応じて認証手法を適応させるシステムである．既存研究では位置情報を GPS により取得しているため，大きな位置誤差が発生する点で課題がある．そのため，本研究では，位置情報に WiFi RTT[9] による位置情報を用いることにより，認証の高精度化，高利便性の実現を目標とする．本研究の認証環境として，研究室に所属している学生における研究室内リソースアクセスを設定した．

本研究では実験を 2 つ実施した，実験 1 では WiFi RTT を利用した位置情報の実現可能性，精度について評価を行った．実験 2 では，WiFi RTT による位置情報を補助する背景情報である，予定管理ツールを利用した時間情報，ユーザーがブラウザに設定する言語情報について，実現可能性，システムの利便性の面で評価を行った．本研究のシステムフローを以下の図 2 に示す．

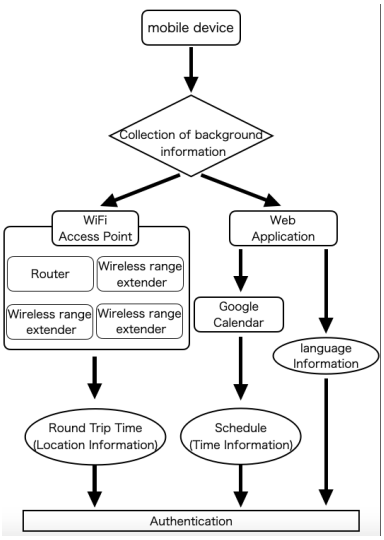


図 2 Systemflow

#### 5.1.1 位置情報：Wi-Fi RTT

WiFi RTT は無線 LAN を利用して RTT(応答時間) を測定し，それにより位置の測定を行う．この距離測位は WiFi アクセスポイントとスマートフォンなどの WiFi 受信端末があれば可能である．WiFi RTT による距離測位は，一度のリクエスト中において，端末とアクセスポイント間での通信を複数回行い，その応答速度の平均値により算出される．この時に 3 つ以上のアクセスポイントを利用することで，1 ~ 2 メートルの誤差で位置を測定することが可能であると言われている [9]．例えば，ユーザー 1, 2, 3 はそれぞれ 7000mm，5000mm，6000mm の RTT を取得する．この状況下で，ユーザー 1 が保有する RTT である 7000mm 地点からユーザー 2 によるアクセスが行われた場合，そのアクセスは怪しいと判断することができる．

#### 5.1.2 時間情報：研究室で利用されている予定管理ツール

研究室で利用されている予定管理ツールによる時間情報を用いることにより，ユーザーが研究室にいる可能性が高い時間を滞在時間要素として，いない可能性が高い時間を不滞在時間要素として利用することができる．既存研究では不滞在時間要素の確認が不可能なので，信頼度が曖昧なユーザーに対して，高いレベルの認証を要求している．しかし，本研究では不滞在時間要素の確認をすることで，明

らかに信頼度が低いユーザーに対して高いレベルの認証を要求できる．これは積極的な判断であり，既存研究より強力な認証だと言える．

本研究では，予定管理ツールとして Google カレンダーを使用した．図 3 に Google カレンダーを時間情報として用いた際の例を示す．ユーザーが研究室に滞在している午前 10 時 ～ 12 時の間に行われるアクセスは，Google カレンダーによる滞在時間要素を確認することで，正当なアクセスである可能性が高いと判断することができる．午後 1 時 ～ 2 時 30 分の間に行われるアクセスについては，Google カレンダーによる不滞在時間要素を確認することにより，不当なアクセスである可能性が高いと判断することができる．

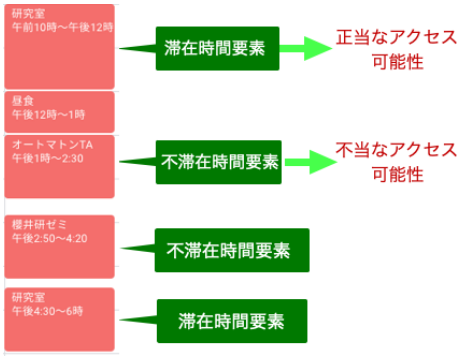


図 3　Google カレンダー情報：使用例

#### 5.1.3　言語情報：ユーザーのブラウザ設定言語

ユーザーがブラウザに設定している言語情報を用いることで，アクセスを行うユーザーの本人性の信頼度合いを予測することができる．言語情報は，HTTP ヘッダ内の html 文を抽出することで，ユーザーがブラウザに設定している言語を把握することができ，この手法により取得することができる．

## 6.　実験結果

### 6.1　実験 1

実験 1 では，WiFi RTT を位置情報として利用する場合の実現可能性と精度について評価を行う．

#### 6.1.1　評価 1：実現可能性

WiFi RTT による情報は，WiFi ルーター等の WiFi アクセスポイントとスマートフォン等の WiFi 受信端末があれば取得することができる．この機能は IEEE802.11mc プロトコルをベースとしており，アプリが使用する位置情報の精度を向上している [10]．WiFi RTT を使用するための API は，Android Developers で公開されている [11]．本研究では，WiFi ルーターには Google Nest WiFi ルーターを一台，拡張ポイントには Google Nest WiFi 拡張ポイントを 3 台利用し，WiFi 受信端末には Google Pixel 3a を利用した．これらの機器を用いることにより，WiFi RTT による位置情報を取得することができた．Android Studio における Logcat の出力を以下の図 4 に示す．図 4 で，「櫻井研究室」という名前のルーターがスキャンできている事が確認できる．図 5 では拡張ポイントのスキャンが成功した事が確認できる．図 6 では，ルーターからの RTT が mm 単位で取得できている事が確認できる．これらのことから，WiFi RTT による位置情報は実現可能である．

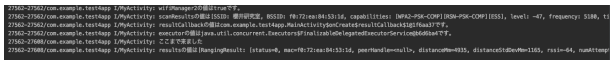

図 4　Android Studio における Logcat の出力

図 5　拡張ポイントのスキャン

図 6　RTT 情報の取得

#### 6.1.2　評価 2：精度

本研究では，研究室内において，学生が作業を行うスペースをアクセスポイントで囲み，その範囲内の 9 分割点を観測地点とした．観測地点それぞれの箇所で，10 回ずつ WiFi RTT を取得した．図 7 はその実際の図であり，図中に記載されている数値の単位はセンチメートル，丸は WiFi アクセスポイントの設置地点，バツの地点が観測地点である．観測地点 1 において取得した WiFi RTT の取得結果の一部を図 8 に示す．図 8 の取得結果より，背景情報としてアクセスポイント情報とそこからの距離を抽出した．

抽出した距離に基づいて測位を行った．観測点をの座標を $(x, y)$，アクセスポイントの座標をそれぞれ $(x_a, y_a)$, $(x_b, y_b)$, $(x_c, y_c)$, $(x_d, y_d)$ とすると，距離情報に基づいた以下の円の方程式が $a, b, c, d$ についてたてられる．

$$(x - x_a)^2 + (y - y_a)^2 = {r_a}^2 \tag{1}$$

このうち 2 つの円の方程式の差を取ることで以下の様な直線の式が 3 つたてられる．

$$2(x_b - x_a)x + {x_a}^2 + 2(y_b - y_a)y + {y_a}^2 - {y_b}^2 = {r_a}^2 + {r_b}^2 \tag{2}$$

この 3 つの直線の方程式から，それぞれの交点を求め，その三つの交点を頂点とする三角形の中点を計算し，それを推定端末位置としている．この手法による位置推定は，Google I/O '18 において，WiFi RTT を用いた測位として Google の技術者により紹介された手法である [12], [13]．最後に，推定した端末の位置と正しい位置座標間の距離を計算することで，誤差を求めた．観測点 1~9 における位置推定の誤差を以下の図 9 ～ 17 に示す．また，各観測地点の誤差の平均値を図 18 に示す．これらの結果から，最も誤差が小さく

精度が良い観測点は，誤差平均 365.6497649401748mm の観測点 1 であり，最も誤差が大きく精度の悪い観測点は，誤差平均 4532.422314593179mm の観測点 2 であった．最も精度の良い観測点 1 では，WiFi RTT の理想的な誤差範囲である 1~2m の範囲内であるが，最も精度の悪い観測点 2 では，誤差が 1~2m の範囲外であった．

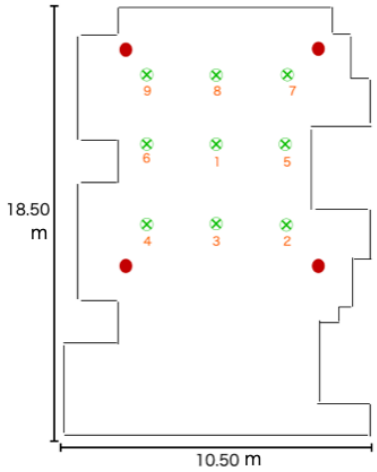


**図 7** 研究室内の 9 区分化

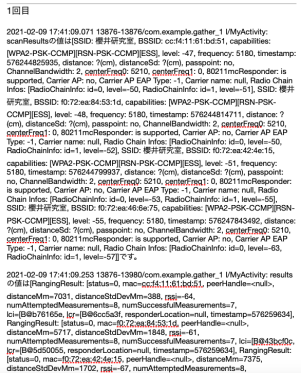


**図 8** WiFi RTT の取得結果 (一部)

Error[0](distance): 556.4770194319176 [mm]
Error[2](distance): 303.60480891491676 [mm]
Error[3](distance): 85.44675419711817 [mm]
Error[4](distance): 491.98864840383004 [mm]
Error[5](distance): 311.3067348046259 [mm]
Error[7](distance): 354.45470391675514 [mm]
Error[9](distance): 456.2696845821343 [mm]

**図 9** 観測点 1 における位置推定の誤差

Error[0](distance): 4179.654133148696 [mm]
Error[2](distance): 4361.70597509931 [mm]
Error[3](distance): 4583.248048058902 [mm]
Error[4](distance): 4160.035042112325 [mm]
Error[5](distance): 4344.517823894396 [mm]
Error[7](distance): 5006.44772649663 [mm]
Error[9](distance): 5091.354696595005 [mm]

**図 10** 観測点 2 における位置推定の誤差

Error[0](distance): 1341.199571745925 [mm]
Error[1](distance): 1685.3978494806433 [mm]
Error[2](distance): 1556.4687282489194 [mm]
Error[3](distance): 1715.7820859377734 [mm]
Error[4](distance): 1320.4498357983796 [mm]
Error[6](distance): 1637.10034894723 [mm]
Error[7](distance): 3150.7414203409153 [mm]
Error[8](distance): 1164.0890421649733 [mm]
Error[9](distance): 1416.8618452381359 [mm]

**図 11** 観測点 3 における位置推定の誤差

Error[0](distance): 2860.405880767342 [mm]
Error[1](distance): 2590.795453341714 [mm]
Error[2](distance): 1743.386954224912 [mm]
Error[3](distance): 5070.322034847257 [mm]
Error[4](distance): 1837.9440903188809 [mm]
Error[6](distance): 2004.961512278507 [mm]
Error[7](distance): 1780.6770737021293 [mm]
Error[8](distance): 653.681030098626 [mm]
Error[9](distance): 3048.338538190422 [mm]

**図 12** 観測点 4 における位置推定の誤差

Error[0](distance): 1243.8030655051707 [mm]
Error[1](distance): 1108.2948108804655 [mm]
Error[3](distance): 463.99534839412684 [mm]
Error[4](distance): 2434.695729261369 [mm]
Error[5](distance): 1016.5332123435159 [mm]
Error[6](distance): 1175.548862651182 [mm]
Error[7](distance): 3456.529602819268 [mm]
Error[9](distance): 75.50531753806698 [mm]

**図 13** 観測点 5 における位置推定の誤差

Error[0](distance): 538.8582698918212 [mm]
Error[2](distance): 573.6997836809018 [mm]
Error[3](distance): 665.8437193060723 [mm]
Error[4](distance): 538.2596221173641 [mm]
Error[6](distance): 549.0612509789809 [mm]
Error[7](distance): 479.31234889543646 [mm]
Error[9](distance): 813.2055994029172 [mm]

**図 14** 観測点 6 における位置推定の誤差

Error[0](distance): 3058.5171340683796 [mm]
Error[1](distance): 4034.430681071104 [mm]
Error[3](distance): 3373.7793327737595 [mm]
Error[5](distance): 3291.7768671205217 [mm]
Error[6](distance): 2053.826350760058 [mm]
Error[7](distance): 3304.2400435610725 [mm]
Error[8](distance): 2778.2735199972963 [mm]
Error[9](distance): 3347.7824113043293 [mm]

**図 15** 観測点 7 における位置推定の誤差

Error[1](distance): 1317.8398695819492 [mm]
Error[2](distance): 1574.8828730725309 [mm]
Error[3](distance): 1932.2130726162345 [mm]
Error[4](distance): 6114.417047731581 [mm]
Error[5](distance): 4141.914583739701 [mm]
Error[7](distance): 3757.464357758754 [mm]
Error[9](distance): 2637.741309161373 [mm]

**図 16** 観測点 8 における位置推定の誤差

Error[0](distance): 2632.944248933373 [mm]
Error[2](distance): 1581.3855175357073 [mm]
Error[3](distance): 1792.0905182139104 [mm]
Error[4](distance): 1898.4902955306088 [mm]
Error[5](distance): 1826.8730050356087 [mm]
Error[6](distance): 5551.618040873986 [mm]
Error[7](distance): 2813.010778470973 [mm]
Error[8](distance): 2345.846628483123 [mm]
Error[9](distance): 2607.618685502886 [mm]

**図 17** 観測点 9 における位置推定の誤差

観測点1：365.6497649401748
観測点2：4532.422314593179
観測点3：1665.343414211432
観測点4：2398.945840863309
観測点5：1371.863243674145
観測点6：594.0343706104992
観測点7：3155.328292582065
観測点8：3068.067587666017
観測点9：2561.097524286686

図 18　各観測地点における誤差の平均

### 6.2　実験 2

実験 2 では，WiFi RTT による位置情報を補助するための要素として用いる，予定管理ツールによる時間情報，ユーザーのブラウザ設定言語による時間情報の実現可能性，利便性について評価を行う．

#### 6.2.1　評価 1：実現可能性

ユーザーの予定管理ツールによる時間情報は，Google カレンダーを利用することにより取得する事ができる．Google カレンダーを参照する API は公開されているため，本研究ではユーザーが設定している Google カレンダーにおけるスケジュールを参照するアプリケーションを作成した．Google カレンダー情報を取得した実際の図を以下の図 19 に示す．図 19 の上部赤枠内で，Google カレンダー情報を html 文で取得することに成功している事が確認できる．

ユーザーの言語情報は，ユーザーがブラウザに設定している言語を HTTP ヘッダの html を取得することにより確認する．言語情報を確認するためのアプリケーションは，時間情報を取得するアプリケーションと統合している．図 19 中の赤枠内中央の箇所から，ユーザーの使用言語を html で取得できている事が確認できる．これらのことから，予定管理ツールによる時間情報，ブラウザ設定言語による時間情報は，実現可能である．

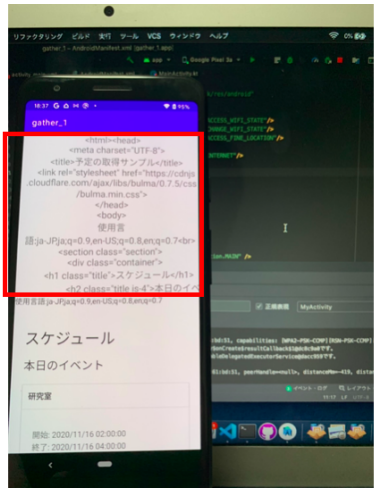


図 19　時間情報，言語情報の取得

#### 6.2.2　評価 2：利便性

本実験で追加した機器は，WiFiRTT を取得するために用いた GoogleNestWiFi ルーターが 1 機，GoogleNestWiFi 拡張ポイントが 3 機であり，これらの機器は市販され簡単に入手可能であり，WiFiRTT 情報の収集以外に一般的な WiFi ルーターとして使用可能である．また，本研究によるユーザーの負担は，本研究で用いた 3 つの背景情報を取得するためのアプリケーションの導入，予定管理ツールへのスケジュールの記入である．しかし，実際にユーザーが認証を行う際には，本研究で収集した 3 つの背景情報を入力する必要はない．これらのことから，提案システムの利便性は高いと判断できる．

## 7.　考察

### 7.1　考察 1：WiFi RTT による位置情報

本研究の実験結果から，適応型認証の背景情報として WiFi RTT による位置情報が利用できることが判明した．しかし，推定位置の誤差は観測点によってばらつきがあり，最も精度の高い観測点 1 で誤差が約 36.6cm，最も精度の低い観測点 2 で約 4.53m の誤差が発生した．誤差が大きく出た原因として，人の手によるアクセスポイントの設置，アクセスポイントや端末座標の測定による偶然誤差が影響していると考えることができる．また，図 7 から分かる様に，研究室内の柱などによる歪な形状が，端末とアクセスポイント間の応答時間に誤差を生じていると考えられる．また，大学内の多数のアクセスポイントからの信号が端末に影響を与えて，応答時間に影響を与えていることも考えられる．しかし，最も精度の高い箇所では，数センチの誤差しか発生しておらず，WiFi RTT による位置情報だけでなく，他の背景情報と組み合わせて認証子として用いることで，十分効果的だと考えることができる．また，WiFi RTT は現在のところ，その情報の取得のために利用可能な WiFi アクセスポイントは，Google 社の製品が一般的である．これから，WiFi RTT による位置情報の取得に対応できる製品は広まっていくことが予想される．その場合は，自分の研究室以外のアクセスポイントからも WiFi RTT による位置情報を取得することが可能になり，研究室内に多くのアクセスポイントを設置せずとも，より高精度な測位結果が得られる様になると期待できる．

### 7.2　考察 2：ユーザー依存情報

本研究の適応型認証の背景情報として用いている言語情報はユーザーがブラウザに設定してある言語であり，時間情報は予定管理ツール情報を用いている．これら背景情報の精度はユーザーに依存しており，どのようなユーザーがどれだけの粒度でその情報を入力するかで認証の精度に影響する．例えば研究室内における学生が大学院生の場合，母国の言語にかかわらず，PC 等の標準言語を英語に設定

する場合がある．このような場合は，言語情報が「英語」の段階で，「大学院生」という属性として判定することを一考する必要がある．また，予定管理ツール情報については，ユーザーそれぞれで入力を行う場合，ユーザー毎の予定記載粒度の差から認証の背景情報として使うことが難しくなる場合がある．その対策の為に，個人でのカレンダー入力だけでなく，研究室や研究グループ単位での予定などがあれば，メンバー間での予定記載粒度の差が縮まることが予想される為，対策を取ることができる．これらのことを考慮することで，WiFiRTT による位置情報を補助する要素として十分な効果が期待できる．

### 7.3 一般化

本研究では認証環境として，「大学の研究室に所属する学生における研究室内リソースアクセス」を設定した．この認証環境を他の環境で考える．例えば認証の特定状況として「オフィス内」を設定する場合は，入退社の際に用いるICカード，新しいスマートフォン端末を社員それぞれに与えている場合は，気圧計情報を使ってオフィスの階層をある程度推定することが可能と考えられる．また，部屋の構造や階層などによっては，日時やその日の天気情報等から太陽の照度がある程度予想されるので，その情報も補助情報として利用できる可能性がある．いずれの環境であっても，近年のモバイル端末のセンサーの充実度を活用することで，新たな機材の導入を必要とせずに，アプリケーションだけで適応型認証の精度向上が望める．

## 8. おわりに

本研究では，WiFiRTT による位置情報を用いた適応型認証の高精度化，高利便性を目標とした．認証環境として「大学の研究室に所属する学生における，研究室内リソースアクセス」を設定した．この設定による実験結果から，WiFi RTT 情報による位置測位だけでは精度にばらつきが存在した．精度の高い点を考えた場合，十分適応型認証の背景情報として用いることができるが，精度の低い箇所では，それのみを用いた認証では十分と言える精度ではなかった．しかし，認証環境によって得られる他の背景情報と組み合わせることで，適応型認証は高精度化は望むことができ，背景情報の組み合わせによる認証の一要素としては，十分効果が期待できることが判明した．また，近年は認証システムの利便性が重要視されている．利便性の評価要件の 1 つに，「効率よく使用できる」という点があるので，その面での利便性の高さも，特定条件下における適応型認証には十分期待することが可能である．

### 参考文献

[1] 個人情報流出はなぜ止められないのか：入手先 ⟨https://global.jiran.com/jp/article/356⟩（参照 2021-1-31）
[2] ファイアウォールとは—セキュリティ基礎知識・仕組み・種類を初心者向けに解説：入手先 ⟨https://boxil.jp/mag/a1288⟩（参照 2021-1-31）
[3] Patricia Arias-Cabarcos and Christian Krupitzer and Christian Becker.:*A Survey on Adaptive Authentication*, ACM Computing Surveys, (2019).
[4] Reza Fathi and Mohsen Amini Salehi and and Ernst L. Leiss.:*User-friendly and secure architecture (UFSA) for authentication of cloud services*, In Proceedings of the IEEE International Conference on Cloud Computing (CLOUD ’ 15), 516-523, (2015).
[5] Davy Preuveneers and Wouter Joosen.:*SmartAuth: Dynamic context fingerprinting for continuous user authentication*, In Proceedings of the ACM Special Interest Group on Applied Computing (SIGAPP ’ 15),2185-2191, (2015).
[6] Zhan Liu and Riccardo Bonazzi and Yves Pigneur.:*Privacy-based adaptive context-aware authentication system for personal mobile devices*, J. Mob. Multimed. 12, 1-2 (Apr. 2016),159-180,(2016).
[7] David Freeman and Sakshi Jain and Markus Durmuth and Battista Biggio and Giorgio Giacinto.:*Who are you? A statistical approach to measuring user authenticity*, In Proceedings of the Network and Dis- tributed System Security Symposium (NDSS’16), 1-15, (2016).
[8] GPS による位置誤差：入手先 ⟨http://www.ne.jp/asahi/nature/kuro/HGPS/principle_gps.htm⟩（参照 2021-1-31）
[9] Wi-Fi RTT：入手先 ⟨ https://developer.android.com/guide/topics/con rtt?hl=ja ⟩（参照 2021-1-31）
[10] WiFi RTT (802.11mc)：入手先 ⟨https://source.android.com/devices/tech/connect/wifi-rtt?hl=ja⟩（参照 2021-2-11）
[11] Wi-Fi location: ranging with RTT Android Developers：入手先 ⟨https://developer.android.com/guide/topics/connectivity/wifi-rtt⟩（参照 2021-2-11）
[12] How to get one-meter location-accuracy from Android devices (Google I/O ’18)：入手先 ⟨ https://www.youtube.com/watch?v=vywGg5rGODU ⟩（参照 2021-2-10）
[13] Wi-Fi RTT による屋内測位アプリを作ろう：入手先 ⟨ https://speakerdeck.com/napplecomputer/wi-fi-rttniyoruwu-nei-ce-wei-apuriwozuo-rou ⟩（参照 2021-2-10）