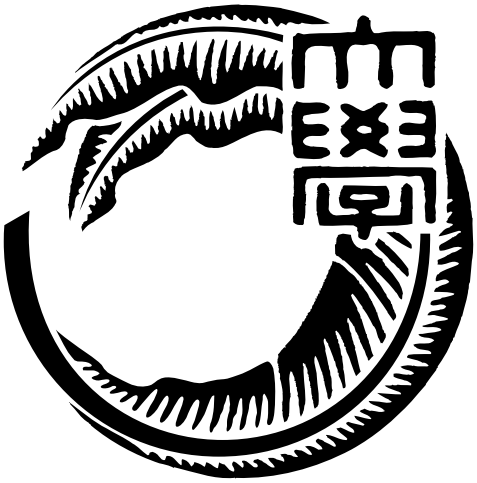
令和元年度 卒業論文

公開暗号方式を用いたSSH認証による アカウント作成の簡略化

Simplify Creating Account by SSH Authentication using Public-key Cryptography



琉球大学工学部情報工学科

165714D 与那嶺東

指導教員 長田智和,谷口祐治

目 次

図 目 次

表 目 次

第1章　はじめに

1.1　背景と目的

1.2　論文の構成

第2章　基礎概念

2.1

2.2

第3章　提案手法

3.1　ボツ案

3.1.1　ログイン状態のセッション橋渡し

実装したいこと
公開鍵暗号方式による ssh 認証を成功した際に，WEB サービスでログイン状態になる．ここでいう，ログイン状態は，セッションを用いて，ステートフルな通信をする (HTTP 通信はステートレスな通信).

実装するための手段案
ログイン状態のセッション情報を橋渡しすることにより，実現しようと考案した。以下の図 3.1 を用いながら説明する。

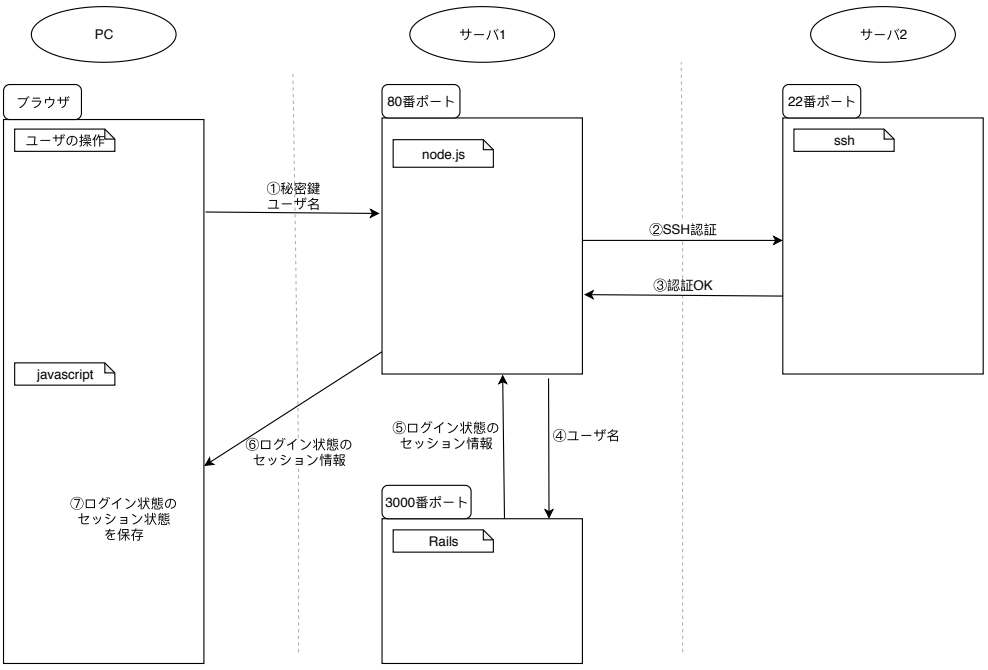


図 3.1: ログイン状態のセッション橋渡し案

図 3.1 の① 〜 ③では，エンドユーザはブラウザを用いて，公開鍵暗号方式による ssh をしている。図 3.1 の④では，HTTP リクエストの POST をしている。図 3.1 の⑤では，HTTP レスポンスが来て，そのヘッダ情報に「セッション情報を Cookie に保存する」という情報が付与されている (のちのち, アクセス制限で,localhost からしかアクセスできないようにする). 図 3.1 の⑥では，図 3.1 の⑤の情報を, ブラウザの javascript 側に socket 通信で渡す. 最後に，図 3.1 の⑦で，ログイン状態のセッション情報を Cookie に保存する.

ブラウザを用いての検証
ログイン中のセッション情報を, ブラウザの Cookie に保存することで，ログインすることが可能かを検証する。
検証環境
- 機器
  - MacBook Pro (Retina, 13-inch, Early 2015)
  - macOS High Sierra(バージョン 10.13.6)
- サーバ側
  - Ruby on Rails(6.0.2.1)
  - localhost:3000
- ユーザ側
  - 2っのブラウザ
    - Google Chrome
    - Google Chrome Canary

Crome の開発環境 (デベロッパーツール) を用いて以下の検証を行った。
まず，ログインした際の動きとして，HTTP レスポンスのヘッダ情報に,set-cookie がある.Crome のデベロッパーツールでは，以下の図 3.2 と表示される.

図 3.2: ログイン成功した際の,HTTP レスポンスの抜粋

次に，図 3.2 の HTTP レスポンスを元に，ブラウザの Cookie にセッション情報を保存する。Crome のデベロッパーツールでは，以下の図 3.3 と表示される.
上記の 3.2,3.3 は,HTTP レスポンスのヘッダ情報で, ブラウザの Cookie に値を保存して

図 3.3: ブラウザの Cookie のセッション情報

いる. その後,HTTP リクエストで,Cookie 情報を付与 [1] することにより, ステートレスな

プロトコルである HTTP 上で，状態管理ができる [2](ログイン状態の維持).
現状で以下の図 3.4 のように，ログインしているブラウザ，ログインしていないブラウザがある.

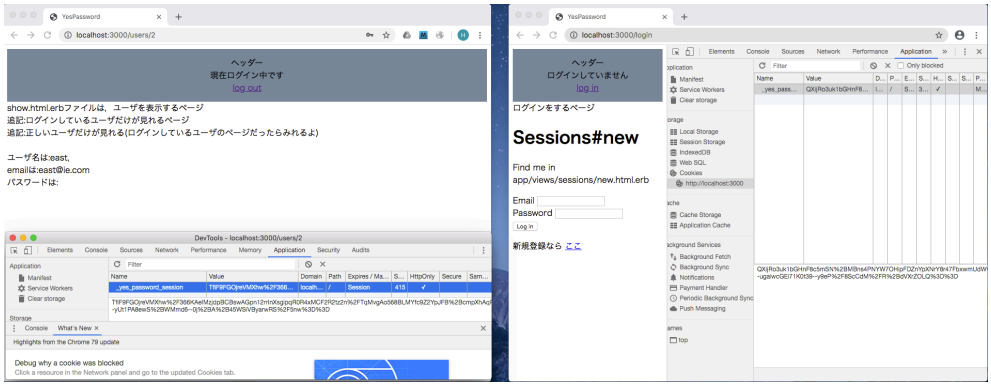


図 3.4: 2っのブラウザ (ログイン状態, ログインしていない状態)

上記の図 3.4 の状態から次の操作を行う. デベロッパーツールを用いて，ログインしているブラウザから，ログインしていないブラウザに，Cookie 情報の, コピーアンドペーストを行い. リロードする. その際の，状態が以下の図 3.5 になり, ログイン状態のセッションを橋渡し (Cookie に保存) することにより, ログインすることができることがわかる.

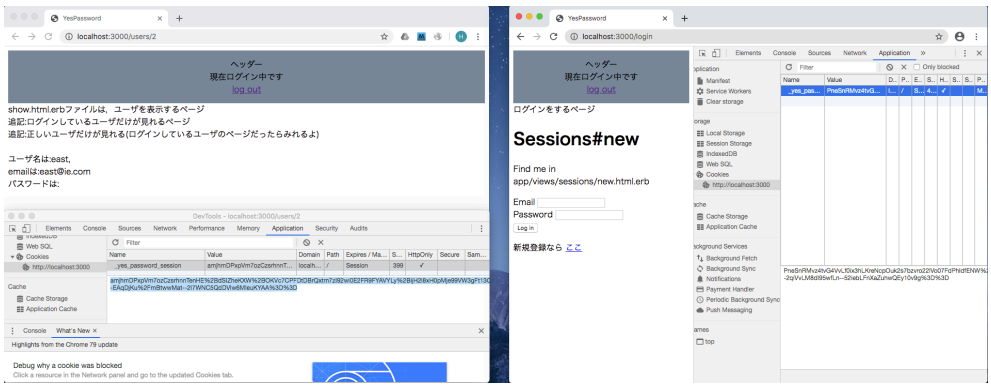


図 3.5: 2っのブラウザ (ログイン状態, ログイン状態)

図 3.5 の補足.
HTTP リクエストをする際に, 新しい Cookie 情報が付与されることが確認できた. そのことにより, ログイン中の 2っのブラウザが別のセッションになっている.

実際に実装
実際に実装を進めて，図 3.1 の① ～ ⑥までできた. しかしながら，図 3.1 の⑦ができなかった. その理由を以下の図 3.6 を元に説明する.
図 3.6 の⑤ で,rails ではセッションを発行する際に,http-only 属性を付与している [3]. その

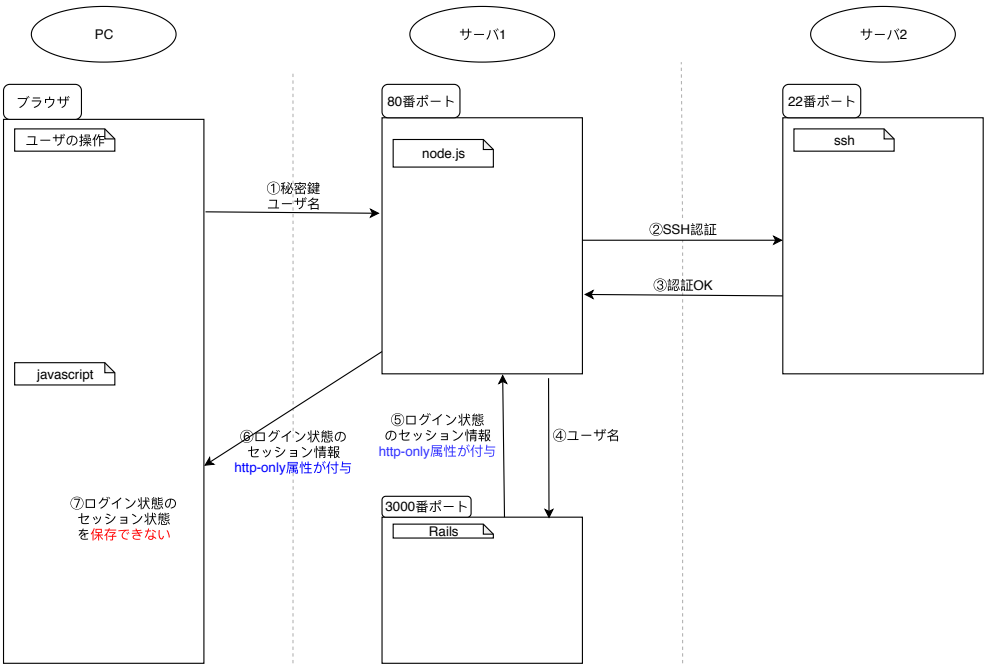


図 3.6: ログイン状態のセッション橋渡し案がボツになった

ことは，ログイン時の HTTP レスポンスである図 3.2 でも現れている．http-only 属性がついた Cookie は,javascript で扱うことができなくなり，セッションハイジャックの対策を行っている [4]．そのため, 図 3.6 の⑦のように, セッション状態を保存することができなかったため,「ログイン状態のセッション橋渡し」の案はボツになった.

第4章　検証

4.1　検証1

4.1.1　検証背景

検証目的

password 認証方式の，新規登録・認証の面倒さを解決するために，第 3 章の提案手法で password 認証方式に変わる，公開鍵暗号方式による ssh 認証を用いた,WEB サービス認証の提案を行った。
しかしながら，提案だけだと，新規登録・認証の面倒さを解決していることの根拠に乏しい。よって，検証を行い，第 3 章の提案手法は”面倒さの軽減”に効果的に繋がっているかの確認をする。

検証手段

検証の手段としては，実際に, 以下の 2っの認証方式の登録・認証を被験者に体験してもらう。

- パスワード方式認証 (以後 ”パスワード認証”と記述する)
- 公開鍵暗号暗号方式による ssh 認証 (以後 ”鍵認証 ”と記述する)

その後，2っの観点から，”面倒さ”を数値化する。
1っ目の観点は「時間」である。”面倒さ”をアカウント登録・認証にかかる時間と推測し，計測化する。詳しい詳細については，検証環境のマニュアルに記述する。
2っ目の観点は「アンケート」である。アンケートには，点数で答える方式, 文字で記入する欄 の 2っがあり，点数で答える方式により”面倒さ ”を数値化する。アンケートの細かい内容は，4.1.3 の検証画面に記述する。
また，アンケートには”面倒さ”を数値化する以外にも，以下の 2っの意味を込める。1っ目の意味は次の通りである。アカウント登録・認証にかかる時間 を，”面倒さ”と予想して検証しているが，その予想を確かめる必要がある。アンケートを取ることにより,「アカウント登録・認証にかかる時間」と,「アンケートによる面倒さ」が比例していることを確認することで，予想を確かめることができる。また，被験者の状態も確認することで，面倒と感じるのが，検証自体に対しての面倒さと関係があるのかを確認する。2っ目の意味は次のとおりである。記入欄で，改善点や感じたことの意見をもらうことで，今後の研究に生きるようなアンケートをもらう。

4.1.2 検証環境

検証場所

第 3 章で記述したとおり，検証場所は学科の VM を用いているため，学科のネットワーク内 (有線 LAN,wifi アクセスポイント ie-ryukyu) から，アクセスして検証を行う。

検証の流れ

ここでは, 被験者に行ってもらう，検証の流れを記述する。時間の観点で”面倒さ”を数値化する検証では, 被験者にはパスワード方式, 鍵方式の登録・認証をそれぞれ行ってもらう。その時，被験者は時間を測る。また，再現性を持って，検証を行うためにマニュアルを作成し, マニュアル通りに検証を行う。アンケートの観点で”面倒さ”を数値化する検証では，時間の観点で”面倒さ”を数値化する検証　が終わった直後に行うようにすることで，被験者の思った感情とアンケート結果の差異が少なくなるようにする。

検証画面

ここでは, 被験者に行ったもらう検証画面をのせる。

以下の図 4.1, 図 4.2 図 4.3 は, 第 3 章の提案手法で実現したパスワード方式, 登録・認証を, 実際に被験者に行ってもらった時のブラウザ画面である。図 4.1 は，アカウント登録画面である。図 4.2 は，図 4.1 で登録したアカウントに認証するための画面である。図 4.3 は，図 4.2 で認証成功した後の画面である。

図 4.1: 検証 1_アカウント作成 (パスワード方式)

図 4.2: 検証 1_認証 (パスワード方式)

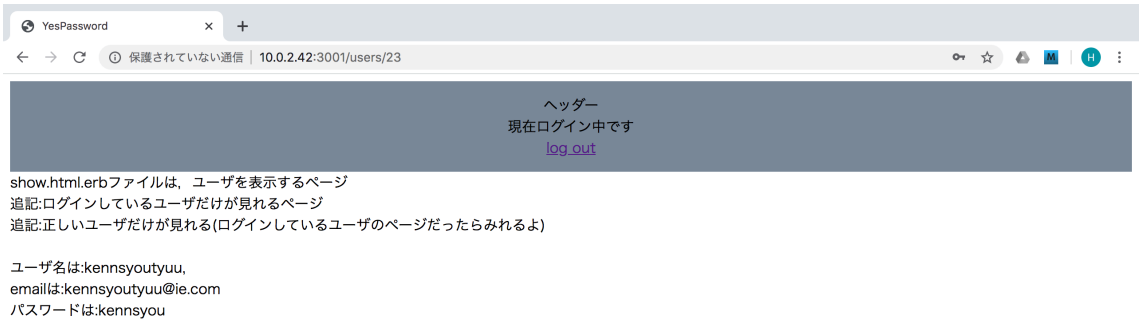


図 4.3: 検証 1_認証成功後 (パスワード方式)

以下の図 4.4, 図 4.5 図 4.6 は, 第 3 章の提案手法で実現した，鍵方式の登録・認証を, 実際に被験者に行ってもらった時のブラウザ画面である。図 4.4 は，アカウント登録画面である。図 4.5 は，図 4.4 で登録したアカウントに認証するための画面である。図 4.6 は，図 4.5 で認証成功した後の画面である。

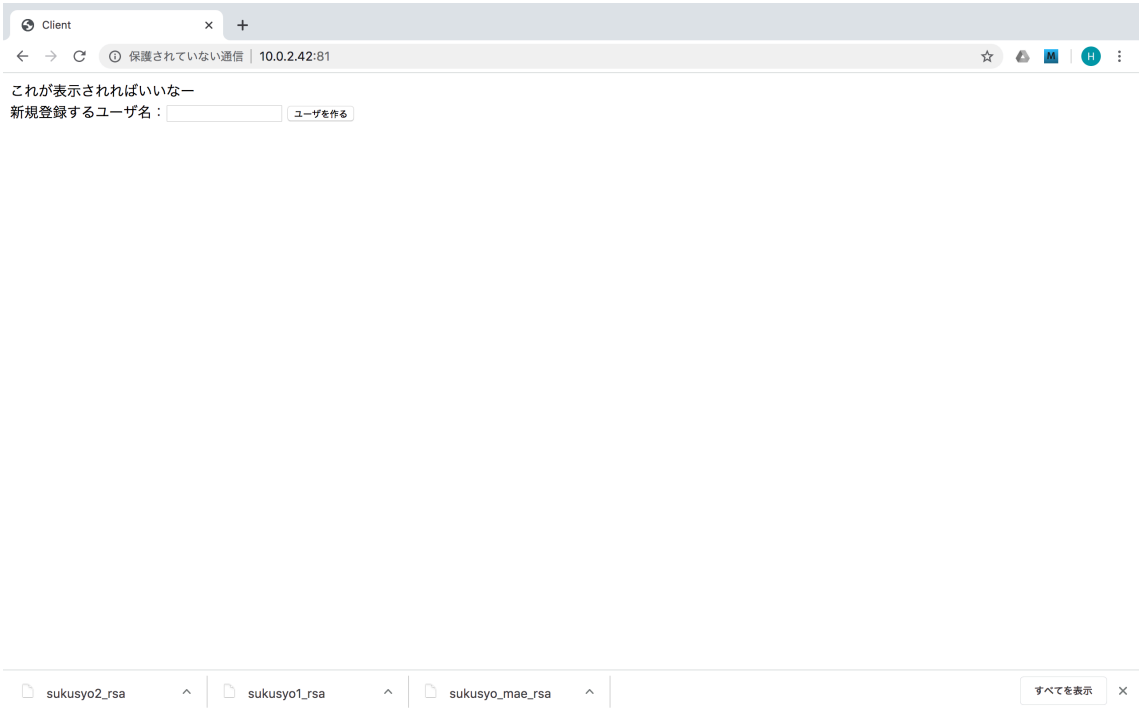


図 4.4: 検証 1_アカウント作成 (鍵方式)

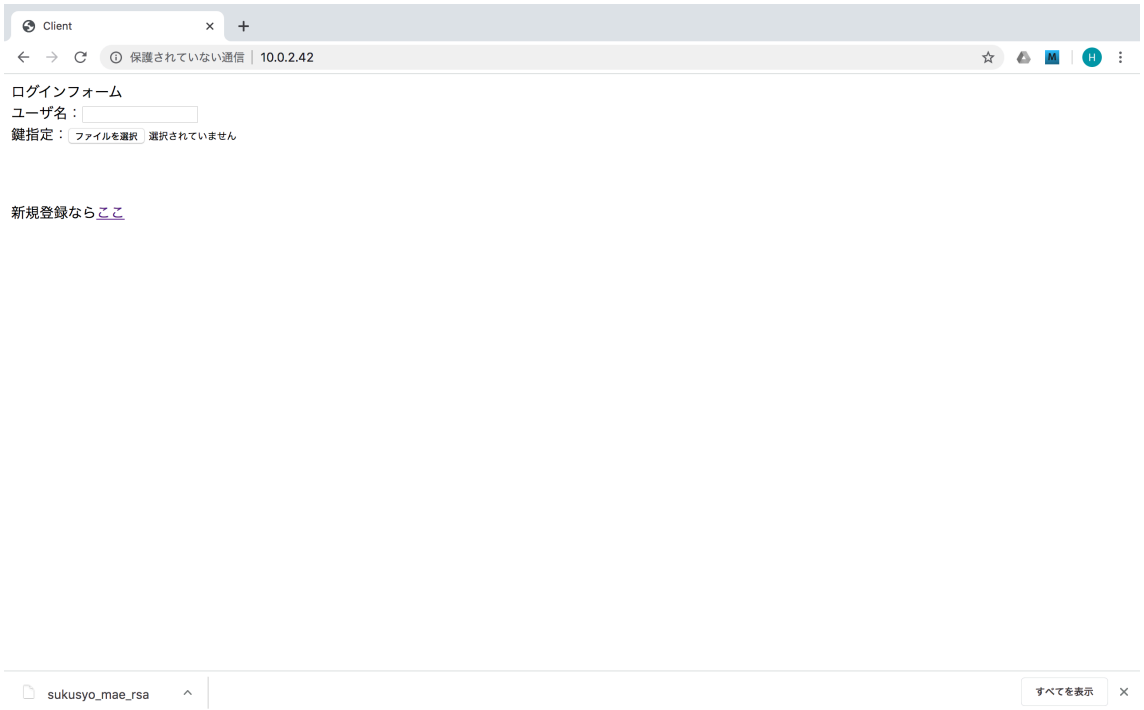


図 4.5: 検証 1_認証 (鍵方式)

図 4.6: 検証 1_認証成功後 (鍵方式)

以下の図 4.7, 図 4.8, 図 4.9, 図 4.10 は図 4.1 〜 図 4.6 までの, 時間計測が終わった直後に行う，アンケートの画面である。図 4.7 では,”検証自体の面倒さ”をアンケートで聞いている。その意図として,”面倒”と感じた感情は, 検証役になること自体が面倒だったことに起因していないかを確かめるためである。図 4.8 では, パスワード方式の登録・認証それぞれについて, どのくらい面倒かを質問している。図 4.9 では, 鍵方式の登録・認証それぞれについて, どのくらい面倒かを質問している。図 4.8, 図 4.9 の質問の意図としては, アンケートの観点から”面倒さ”を数値化することである。図 4.10 では，鍵方式の登録・認証について，被験者の技術視点と利用視点から，自由形式で意見を求めている。図 4.10 の意図としては, よりよい物を作るために, 被験者意見を収集し，今後の方針を決めるためである。


研究検証のためのアンケート
検証を受けた状態について
今回の検証受けるに当たって、正直,検証自体が面倒とどのくらい感じましたか？
(面倒ほど高い数値 [0-5段階])
選択
戻る
次へ

図 4.7: アンケート 1

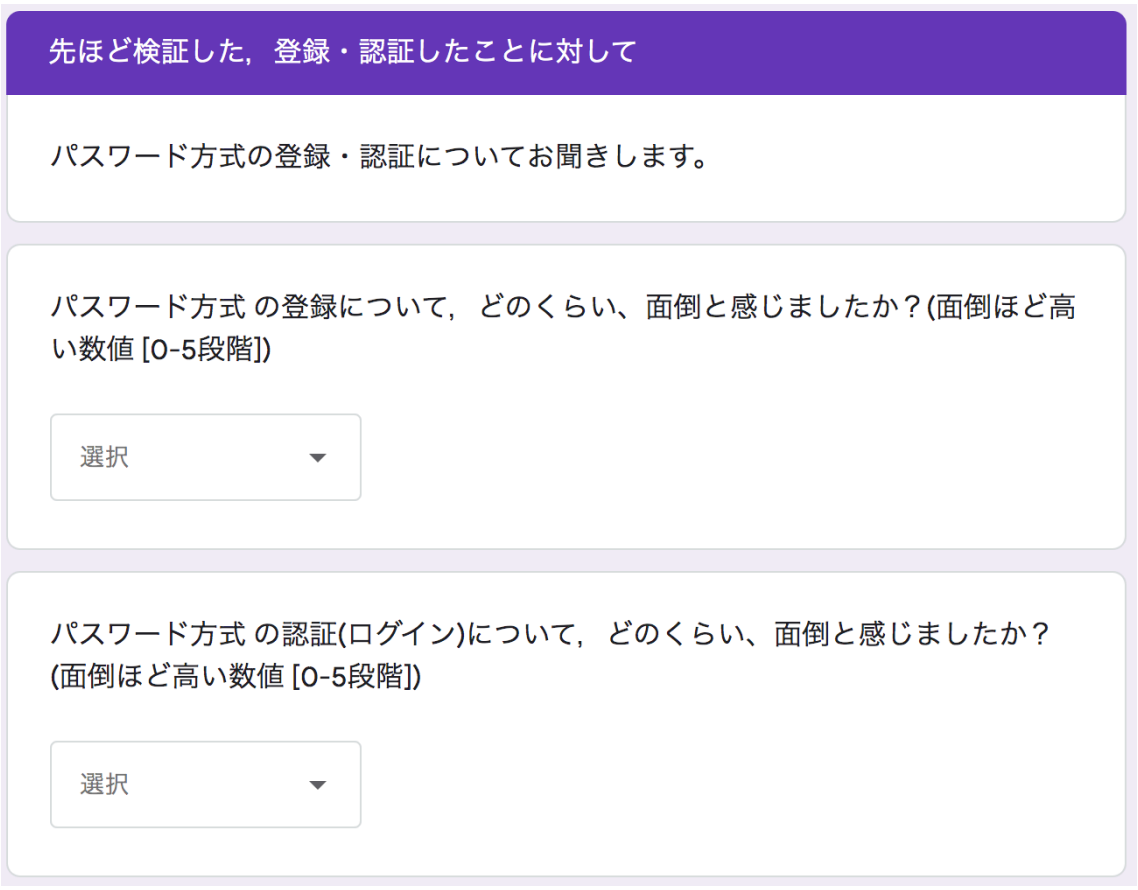

先ほど検証した，登録・認証したことに対して
パスワード方式の登録・認証についてお聞きします。
パスワード方式 の登録について，どのくらい、面倒と感じましたか？(面倒ほど高い数値 [0-5段階])
選択
パスワード方式 の認証(ログイン)について，どのくらい、面倒と感じましたか？(面倒ほど高い数値 [0-5段階])
選択

図 4.8: アンケート 2

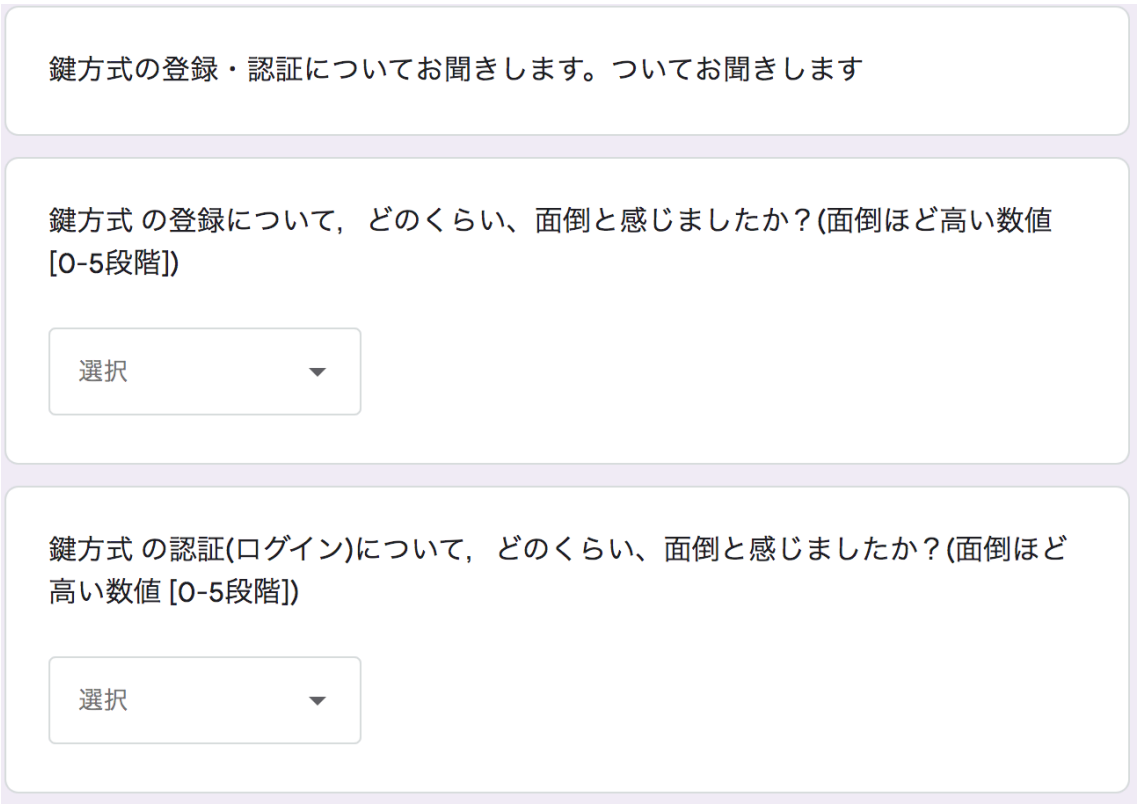


図 4.9: アンケート 3

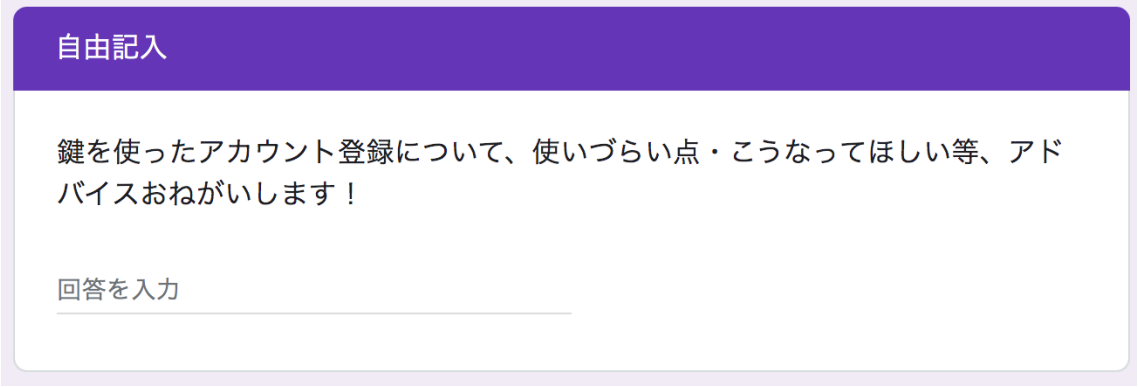


図 4.10: アンケート 4

検証マニュアル

以下の図 4.11, 図 4.12 は, 上記の図 4.1 ～ 図 4.10 の検証を行うためのマニュアルである. マニュアルを作成して, 検証を行った意図としては, 再現性を持って検証を行うためである.

検証方法 (検証目的)，兼マニュアル

時間計測

- 検証環境を準備する。
    - PCを用意する。
        - 被験者のPC もしくは，MacBook Pro (Retina, 13-inch, Early 2015)を用いる
    - ie-ryukyu のアクセスポイントに繋ぐ
    - ブラウザで,以下のURLが開けるか確認する
        - http://10.0.2.42:3001/login
        - http://10.0.2.42:3000/login
- 事前に,検証内容を伝える。

> 今回の実験の被験者になっていただきありがとうございます。被験者に行ってもらう検証次の通りになります。WEBサービスにおける，アカウント登録,認証をしてもらいます。2つの方式の，アカウント登録,認証を行い，時間を計測します。
> 最後に，アンケートを取らせていただくので，協力お願いします。

- 計測方法
    - アカウント登録
        - 被験者は，PCで **サインアップページ** を開いた状態にする

> パスワード方式
>
> > http://10.0.2.42:3001/signup
>
> 鍵方式
>
> > http://10.0.2.42:81/

        - 計測するための確認事項を述べる

> - 「この,webページで，アカウントの登録をお願いします。」
> - 「アカウント登録できたら，報告お願いします。」

        - ストップウォッチで時間を計測する
        - 被験者から報告を聞いたら，ストップウォッチの計測を止める
    - アカウント認証
        - 被験者は，PCで ログインページを開いた状態にする
        - ログアウト状態になっていることを確認する。

図 4.11: 検証マニュアル 1

- ログインんページに行く

  > パスワード方式
  >
  > > http://10.0.2.42:3001/login
  >
  > 鍵方式
  > http://10.0.2.42/

- 計測するための確認事項を述べる
  - 「この,webページで，登録をしたアカウントでログインをお願いします。」
  - 「ログインできたら，報告お願いします。」
- ストップウォッチで時間を計測する
- 被験者から報告を聞いたら，ストップウォッチの計測を止める

アンケート

[アンケート編集場所] ※URL省く
アンケート答える場所

- アカウント登録・認証の検証ありがとうございます。最後に，アンケートを取りたいと思います。
  検証に必要なため，ぜひお願いします。
- アンケート答える場所 で，アンケートに答えてもらう。

被験者から質問来た時に

- 今回の検証受けるに当たって、正直面倒とどのくらいかんじましたか？(0-10段階)

  > (面倒と、思ったことが，検証への意欲と関係あるか確認)

- 登録・認証について，面倒と感じた度合い(10段階)を確認する
  - アンケートにより，「パスワード形式」と「公開鍵暗号方式によるSSH形式」の面倒さを数値化し，比較する。
  - アカウント登録・認証の「面倒さ」と，アカウント登録・認証の「かかる時間」の関係があるかの確認。(アカウント登録・認証にかかる時間 を，面倒さの関係性)
- アンケート

  > 今後の研究に生きるようなアンケートをもらう

図 4.12: 検証マニュアル 2

4.1.3 検証結果

4.1.4 考察

4.2 検証2

第5章　今後の課題

参考文献

[1] https://tools.ietf.org/html/rfc6265#section-1
最終閲覧日:2020/1/20

[2] https://qiita.com/mogulla3/items/189c99c87a0fc827520e
最終閲覧日:2020/1/20

[3] https://qiita.com/yasu/items/8ae3077bdbee606681f6#cookiestore%E3%81%8C%E5%95%8F%E9%A1%8C%E3%81%AA%E3%81%AE%E3%81%8B
最終閲覧日:2020/1/21

[4] https://developer.mozilla.org/ja/docs/Web/HTTP/Cookies
最終閲覧日:2020/1/21

謝辞

本研究の遂行，また本論文の作成にあたり、御多忙にも関わらず終始懇切なる御指導と御教授を賜わりました hoge 助教授に深く感謝したします。

また、本研究の遂行及び本論文の作成にあたり、日頃より終始懇切なる御教授と御指導を賜わりました hoge 教授に心より深く感謝致します。

数々の貴重な御助言と細かな御配慮を戴いた hoge 研究室の hoge 氏に深く感謝致します。

また一年間共に研究を行い、暖かな気遣いと励ましをもって支えてくれた hoge 研究室の hoge 君、hoge 君、hoge さん並びに hoge 研究室の hoge、hoge 君、hoge 君、hoge 君、hoge 君に感謝致します。

最後に、有意義な時間を共に過ごした情報工学科の学友、並びに物心両面で支えてくれた両親に深く感謝致します。

2010 年 3 月

hoge