証明可能安全なパスワード再発行プロトコルついて

大畑 幸矢 †‡　　松田 隆宏 ‡　　松浦 幹太 †

† 東京大学生産技術研究所
153-8505 東京都目黒区駒場 4-6-1
{satsuya,kanta}@iis.u-tokyo.ac.jp

‡ 産業技術総合研究所セキュアシステム研究部門
305-8568 茨城県つくば市梅園 1-1-1
t-matsuda@aist.go.jp

あらまし ID とパスワードを用いたオンラインユーザ認証はこれまでいくつかの問題点が指摘されている一方、その利便性の高さを理由に現在でも頻繁に用いられている認証手法である。人間の記憶力には限度があるためユーザはパスワードを忘れてしまうこともあるが、そのような場合でもサービスを利用できるよう、何らかの方法でパスワードを再発行するためのプロトコルが用意されている場合が多い。本稿では暗号技術研究において用いられる証明可能安全性の考え方に着目し、証明可能安全なパスワード再発行プロトコルについて考察する。

On Provably Secure Password Recovery Protocol

Satsuya Ohata†‡　　Takahiro Matsuda‡　　Kanta Matsuura†

†Institute of Industrial Science, The University of Tokyo
4-6-1 Komaba Meguro-ku, Tokyo 153-8505, Japan
{satsuya,kanta}@iis.u-tokyo.ac.jp

‡Research Institute for Secure System (RISEC),
National Institute of Advanced Industrial Science and Technology (AIST)
1-1-1 Umezono, Tsukuba, Ibaraki, 305-8568, Japan
t-matsuda@aist.go.jp

Abstract Although some problems have been pointed out on a password-based user authentication system, many online services adopt this system because of its usability. Even when a user forget his/her password, he/she can recover it by executing the protocol. In this paper, we consider a provable security treatment for a password recovery protocol.

1 はじめに

1.1 背景

近年、社会の電子化が進み、電子メール、SNS、オンラインでの買い物や銀行取引など、多くのサービスがウェブ上で提供されるようになった。その際に、サーバはサービスを提供する相手が正当なユーザであるかどうかを認証する必要がある。パスワードを用いたオンラインユーザ認証はその利便性の高さを理由に現在でも最も頻繁に用いられているが、いくつかの問題点も指摘されている。そのうちの一つとして、ユーザがパスワードを忘れてしまうことがあるという点が挙げられ、この問題はユーザが人間である以上避けられない。過去にこれを克服するための研究（例えばグラフィカルパスワード [8] や生体認証技術など）が多くなされてきたが、いずれも安全性、完全性、利便性のいずれか（あるいは複数）に問題を抱えており、パスワードを用いた認証方式に取って代わるまでには至っていない。そこで多くのウェブサービスにおいては、パスワードを忘れた場合に対処するプロトコルが用意されている。これは初期登録時にユーザに入力させた個人情報（メールアドレスや携帯電話の SMS、「秘密の質問」など）を用いることによって、そのユーザに（新たな）パ

スワードを再発行するプロトコルである。本稿ではこのプロトコルを「パスワード再発行プロトコル」と呼ぶことにする。

パスワード再発行プロトコルはサービスを利用するユーザにとって便利である反面、安全性上の問題を引き起こす可能性がある。通常のパスワードを用いたユーザ認証や鍵共有などはこれまでに多くの研究が存在し、その安全性についても多数の議論がなされてきている。しかし、仮に通常の認証が安全であったとしても、パスワード再発行プロトコルが脆弱であった場合、悪意を持った攻撃者はその脆弱性を突いて再発行パスワードを入手し、正当なユーザに成りすましてサービスにログインすることが可能となる。このことからパスワード再発行プロトコルに関する研究が非常に重要であることがわかるが、後述するように、そのような既存研究は非常に少ない。

本稿では暗号技術研究において用いられている証明可能安全性の考え方に着目し、証明可能安全なパスワード再発行プロトコルについて考察する。特定の攻撃への対策を考えるのではなく多項式時間攻撃者に対する証明可能安全性を考察することで、攻撃と対策のいたちごっこをある程度防ぐことができ、またセキュリティ理論の体系化につながる。理論体系化の重要性は文献 [9] においても示されている。[1] 具体的には、ID とパスワードを用いてサーバがユーザを正当な利用者かどうかを確認する認証方式に対し、パスワード再発行プロトコルのモデルと安全性を定義し、提案するモデルの下で安全性を証明できる方式を提案する。

モデル設計の困難性　パスワード再発行プロトコルにおいては、ユーザ側がパスワードを忘れている状況において、再発行パスワードを安全にユーザに届けることが求められる。しかし、初期登録時に ID とパスワードのみを登録し、その ID とパスワードを用いて認証を行うシステムにおいては、パスワードを忘れたユーザは秘密情報を一切所持していないため、攻撃者と区別ができない。このことが安全性モデル、及び証明可能安全なプロトコルの設計を困難にしている。

[1] 6 章「情報セキュリティの研究開発における重要分野」の (4) 研究開発の促進基盤の確立と情報セキュリティ理論の体系化

1.2　本論文の貢献

本稿では証明可能安全なパスワード再発行プロトコルについて考察する。前述した困難性により、既存のパスワードに基づく認証方式のモデルや方式を用いて証明可能安全なパスワード再発行プロトコルを構築することは不可能であると思われる。そこで我々は今回、初期登録時にパスワードだけでなくパスワード再発行鍵をユーザに発行し、その再発行鍵を用いてパスワードを再発行するモデルを提案する。再発行鍵は（ハードウェアトークンに入れて保管されるなどして）パスワードとは別の管理がなされ、パスワードを忘れることはあっても再発行鍵を紛失することはないということを前提としている。このようなモデルにおいては、暗号技術研究においても鍵隔離暗号 [1] などに見られる考え方である。また今回提案するモデルは、初期パスワード、及び再発行パスワードをサーバ側が指定し、ユーザが自由にパスワードを設定できない。不便とも感じられるが、パスワードをいったん共有して認証に合格した後は、ユーザ側から自由にパスワードの変更を申し出ることができるため、利便性の観点からはそれほど問題にならないと考えられる。

まず本稿では、パスワード再発行プロトコルのモデル化、及び安全性定義を行う。今回は受動的なりすまし攻撃に対する安全性を取り扱う。その後、それらを満たす方式として ID ベース鍵共有方式とメッセージ認証コード方式を用いた一般的構成法を提案する。また、提案方式の安全性証明を示す。構成要素の ID ベース鍵共有方式とメッセージ認証コード方式に求められる安全性はよく知られたものであり、この一般的構成から数多くの具体的なパスワード再発行プロトコルが構成できる。

1.3　関連研究

パスワード再発行プロトコルを考えた既存研究は非常に少ない。Password Recovery という単語がタイトルに含まれる文献はいくつか存在する（[3] など）が、いずれも本稿で取り扱うパスワード再発行プロトコルに関する研究ではない。Reeder と Schechter による研究 [7, 5] はパスワード再発行プロトコルを含む非常時用の認証手法を Backup authentication、あるいは Secondary authentication と呼び、現在よく用

いられている予備の認証手法を、安全性や利便性の観点などから調査、分析している。しかし、方式提案などを行っているわけではない。

いずれにせよ、パスワード再発行プロトコルの証明可能安全性を考えた研究は、我々の知る限り今のところ存在しない。

1.4 本稿の構成

2 章では後の準備として本稿での表記法、方式の構成において用いられる ID ベース鍵共有とメッセージ認証コードについて、そのモデルと安全性定義を振り返る。3 章ではパスワード再発行プロトコルのモデル、正当性、および安全性を定義する。4 章では方式、及び安全性証明を示す。最後に 5 章で本稿のまとめと今後の課題について述べる。

2 準備

本章では後の準備として表記法の説明、方式の構成において用いられる ID ベース鍵共有とメッセージ認証コードについて振り返る。

2.1 表記法

S が集合ならば、$x \xleftarrow{\$} S$ は S から要素を一様ランダムに取り出し x に代入する操作を表す。$\mathcal{A}$ が確率的アルゴリズムならば、$y \leftarrow \mathcal{A}(x)$ は $\mathcal{A}$ が x を入力として y を出力する操作を表す。なお、アルゴリズム $\mathcal{A}$ が P, Q の二者間でやりとりを伴う場合、$(p_{out}, q_{out}) \leftarrow \mathcal{A}(P(p_{in}) \leftrightarrow Q(q_{in}))$ と書くことで、P, Q はそれぞれ p_{in}, q_{in} を入力し、$\mathcal{A}$ の実行結果としてそれぞれ p_{out}, q_{out} を受け取ることを表す。$x := y$ は x を y と定めることを表す。ϕ は何も出力されないことを表す。k は常にセキュリティパラメータを指すことにする。アルゴリズムの入力における公開パラメータは省略される場合がある。

2.2 ID ベース鍵共有

ID ベース鍵共有 [4, 6] は 3 つのアルゴリズム $(\mathsf{XSetup}, \mathsf{XExt}, \mathsf{XKE})$ からなる。

XSetup: セットアップアルゴリズム XSetup はセキュリティパラメータ 1^k を入力とし、マスター公開鍵 mpk とマスター秘密鍵 msk を出力する。これを $(mpk, msk) \leftarrow \mathsf{XSetup}(1^k)$ と書く。

XExt: 鍵導出アルゴリズム XExt はマスター秘密鍵 msk と ID を入力とし、ユーザ秘密鍵 sk_{ID} を出力する。これを $sk_{ID} \leftarrow \mathsf{XExt}(msk, ID)$ と書く。

XKE: 鍵共有アルゴリズム XKE はユーザ A とユーザ B がそれぞれ自身の秘密鍵 sk_{ID} と相手の ID を入力し、ユーザ A に鍵 K_A、ユーザ B に鍵 K_B を出力する。これを $(K_A, K_B) \leftarrow \mathsf{XKE}(A(ID_B, sk_{ID_A}) \leftrightarrow B(ID_A, sk_{ID_B}))$ と書く。

正当性 いかなる $(mpk, msk) \leftarrow \mathsf{XSetup}(1^k)$ と $sk_{ID_A} \leftarrow \mathsf{XExt}(msk, ID_A)$ 及び $sk_{ID_B} \leftarrow \mathsf{XExt}(msk, ID_B)$ に対しても、$(K_A, K_B) \leftarrow \mathsf{XKE}(A(sk_{ID_A}, ID_B) \leftrightarrow B(sk_{ID_B}, ID_A))$ であれば $K_A = K_B$ となることが正当性として求められる。

安全性 ID ベース鍵共有の安全性を以下のように定義する。ID ベース鍵共有の安全性は、攻撃者 $\mathcal{A}$ とチャレンジャー $\mathcal{B}$ 間の安全性ゲームを用いて定義される。[2]

セットアップ: 攻撃者 $\mathcal{A}$ はサーバの ID である ID_S を選び、チャレンジャー $\mathcal{B}$ に入力する。その後、$\mathcal{B}$ は $(mpk, msk) \leftarrow \mathsf{XSetup}(1^k)$ と $sk_{ID_S} \leftarrow \mathsf{XExt}(msk, ID_S)$ を実行し、マスター公開鍵/秘密鍵のペアとサーバの秘密鍵 sk_{ID_S}、空のリストを生成する。その後、$\mathcal{B}$ は攻撃者 $\mathcal{A}$ に mpk を入力する。$\mathcal{A}$ は以下のクエリを適応的に行うことができる。

開始クエリ Init: $\mathcal{A}$ が ID_i をクエリしてきたときには、$\mathcal{B}$ は $sk_{ID_i} \leftarrow \mathsf{XExt}(msk, ID_i)$ を実行し、リストに (ID_i, sk_{ID_i}) を追加する。$\mathcal{B}$ は $\mathcal{A}$ に何も返さない。また、$\mathcal{A}$ が以下のオラクルに ID をクエリするときには、その ID を事前に Init クエリしておかなければならない。

籠絡クエリ Corrupt: $\mathcal{A}$ が $ID_j (\neq ID_S)$ をクエリしてきた時には、$\mathcal{B}$ はリストを見て sk_{ID_j} を $\mathcal{A}$ に返す。

[2] なお、本稿で定義している ID ベース鍵共有の安全性はパスワード再発行プロトコルの安全性証明を行うために必要最低限のものであり、一時秘密鍵や乱数の漏洩などを考えていない。CK モデルや eCK モデルなどで安全性証明がなされている方式（[2] など）はここで述べているものよりも強い安全性を満たしている。

通信系列クエリ Trans: $\mathcal{A}$ が ID_ℓ をクエリしてきた時には、$\mathcal{B}$ は $(K_\ell, K_S) \leftarrow \mathsf{XKE}(U(ID_S, sk_{ID_\ell}) \leftrightarrow S(ID_\ell, sk_{ID_S}))$ を実行し、その通信系列 $trans$ と (K_ℓ, K_S) を $\mathcal{A}$ に返す。

チャレンジクエリ Test: このクエリは一度だけ行われる。$\mathcal{A}$ が ID^* をクエリした時には、$\mathcal{B}$ は次のように応答する。もし ID^* が過去に Corrupt クエリされたものであれば、$\mathcal{B}$ は $\mathcal{A}$ に $\perp$ を返す。そうでなければ $\mathcal{B}$ は $b \leftarrow \{0,1\}$ をランダムに選ぶ。$b = 0$ であれば $\mathcal{B}$ は $(K_0, K_S) \leftarrow \mathsf{XKE}(U(ID_S, sk_{ID^*}) \leftrightarrow S(ID^*, sk_{ID_S}))$ を実行する。$b = 1$ であれば $\mathcal{B}$ はランダムに K_1 を選ぶ。その後、$\mathcal{B}$ は $\mathcal{A}$ に K_b を返す。

出力: 攻撃者 $\mathcal{A}$ は b の推測値 b' を出力する。$b = b'$ のとき、$\mathcal{A}$ の勝ちとなる。

定義 2.1 全ての多項式時間攻撃者 $\mathcal{A}$ に対し、$\mathsf{Adv}^{\mathsf{IBKX}}_{\mathcal{A}}(k) = 2|\Pr[b = b'] - \frac{1}{2}|$ がセキュリティパラメータ k に対して無視できるとき、ID ベース鍵共有は安全であるという。

2.3 メッセージ認証コード

メッセージ認証コードは 2 つのアルゴリズム $(\mathsf{MMac}, \mathsf{MVer})$ からなる。

MMac: タグ生成アルゴリズム MMac は鍵 $K \in \{0,1\}^k$ とメッセージ m を入力とし、タグ tag を出力する。これを $tag \leftarrow \mathsf{MMac}(K, m)$ と書く。

MVer: 検証アルゴリズム MVer は鍵 $K \in \{0,1\}^k$、メッセージ m、タグ tag を入力とし、$\top$ または $\perp$ を出力する。これを $\top/\perp \leftarrow \mathsf{MVer}(K, m, tag)$ と書く。

正当性 いかなる $K \in \{0,1\}^k$ に対しても $tag \leftarrow \mathsf{MMac}(K, m)$ であれば $\top \leftarrow \mathsf{MVer}(K, m, tag)$ となることが正当性として求められる。

安全性 メッセージ認証コードの適応的選択メッセージ攻撃に対する偽造不可能性は攻撃者 $\mathcal{A}$ とチャレンジャー $\mathcal{B}$ 間の EUF-CMA ゲームによって定義される。

セットアップ: チャレンジャー $\mathcal{B}$ は $K \in \{0,1\}^k$ と空のリストを生成する。その後、攻撃者 $\mathcal{A}$ は以下のクエリを適応的に行うことができる。

タグ生成クエリ TagGen: $\mathcal{A}$ がメッセージ m_i をクエリしてきた時には、$\mathcal{B}$ は $tag_i \leftarrow \mathsf{MMac}(K, m_i)$ を実行する。その後、$\mathcal{B}$ は $\mathcal{A}$ に tag_i を返し、リストに (m_i, tag_i) を追加する。

出力: 攻撃者 $\mathcal{A}$ は (m^*, tag^*) のペアを出力する。m^* がリストに存在せず、かつ $\top \leftarrow \mathsf{MVer}(K, m^*, tag^*)$ であるときに $\mathcal{A}$ の勝ちとなる。

定義 2.2 全ての多項式時間攻撃者 $\mathcal{A}$ に対し、$\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{A}}(k) = \Pr[\mathcal{A}\ wins]$ がセキュリティパラメータ k に対して無視できるとき、メッセージ認証コードは適応的選択メッセージ攻撃に対して偽造不可能性を持つという。

3 パスワード再発行プロトコル

本章ではパスワード再発行プロトコルのモデル、及び安全性定義について述べる。1.2 節で述べたように、パスワードはユーザ側でなくサーバ側で指定するモデルになっている。

3.1 モデル

パスワード再発行プロトコルは 4 つのアルゴリズム $(\mathsf{SSetup}, \mathsf{Gen}, \mathsf{RePGen}, \mathsf{Auth})$ からなる。

SSetup: サーバセットアップアルゴリズム SSetup はセキュリティパラメータ 1^k を入力とし、サーバの公開鍵 pk と秘密鍵 sk を出力する。これを $(pk, sk) \leftarrow \mathsf{SSetup}(1^k)$ と書く。

Gen: パスワードおよびパスワード再発行鍵生成アルゴリズム Gen は秘密鍵 sk と ID を入力とし、パスワード pw とパスワード再発行鍵 rk を出力する。これを $(pw, rk) \leftarrow \mathsf{Gen}(sk, ID)$ と書く。

RePGen: パスワード再発行アルゴリズム RePGen はユーザ U が自身の ID である ID_U とサーバの ID である ID_S、パスワード再発行鍵 rk を入力し、サーバ S が ID_U とパスワード再発行鍵 rk、パスワード pw、秘密鍵 sk を入力することで、ユーザの新たなパスワード pw'_U 及びそれに対応してサーバ側が保持するパスワード pw'_S を出力する。これを $(pw'_U, pw'_S) \leftarrow \mathsf{RePGen}(U(ID_U, ID_S, rk) \leftrightarrow S(ID_U, rk, pw, sk))$ と書く。

Auth: 認証アルゴリズム Auth はユーザ U が ID_U と ID_S、パスワード pw を入力し、サーバ S が ID_U とパスワード pw、秘密鍵 sk を入力することで、サーバ側に $\top$ あるいは $\bot$ を出力する。これを $(\phi, \top/\bot) \leftarrow \mathsf{Auth}(U(ID_U, ID_S, pw) \leftrightarrow S(ID_U, pw, sk))$ と書く。

正当性　正しく生成された鍵においてはユーザとサーバに出力される再発行パスワードが同じものになること、正しく生成されたパスワードにおいては認証に合格すること、及びパスワードが再発行された後は新しいパスワードで認証に合格することを正当性として要求する。すなわち、いかなる $(pk, sk) \leftarrow \mathsf{SSetup}(1^k)$ と $(pw, rk) \leftarrow \mathsf{Gen}(sk, ID)$ に対しても

1. $(pw'_U, pw'_S) \leftarrow \mathsf{RePGen}(U(ID_U, ID_S, rk) \leftrightarrow S(ID_U, rk, pw, sk))$ であれば $pw'_U = pw'_S$ であること
2. (a) $(\phi, \top) \leftarrow \mathsf{Auth}(U(ID_U, ID_S, pw) \leftrightarrow S(ID_U, pw, sk))$ であること
   (b) $(pw, rk) \leftarrow \mathsf{Gen}(sk, ID)$ かつ $(pw'_U, pw'_S) \leftarrow \mathsf{RePGen}(U(ID_U, ID_S, rk) \leftrightarrow S(ID_U, rk, pw, sk))$ であれば $(\phi, \top) \leftarrow \mathsf{Auth}(U(ID_U, ID_S, pw'_U) \leftrightarrow S(ID_U, pw'_S, sk))$ であること

が正当性として求められる。

補足　運用の流れは以下の通りである。サーバは SSetup を走らせて公開鍵と秘密鍵のペアを生成する。ユーザの初期登録時に Gen を用いてユーザごとにパスワード pw と再発行鍵 rk を生成する。ユーザは自身の ID とパスワードを用いて Auth によって自身を認証する。ユーザがパスワードを忘れた際には RePGen を用いて再発行パスワード pw' を生成する。その際に、初期登録時に生成された再発行鍵を用いる。

パスワード再発行プロトコルのモデル、及び安全性定義に関しては既存の ID ベース認証のモデルを拡張したものである。運用上の仮定として、Gen アルゴリズムで生成される初期パスワード pw と再発行鍵 rk は何らかの安全な方法でユーザに配送されることを仮定している。また再発行パスワードが元と同じか異なるかについては、その両者を区別していない。

3.2　安全性定義

パスワード再発行プロトコルにおける受動的なりすまし攻撃に対する安全性 (Security against Impersonation under Passive Attack, SIPA) は、攻撃者 $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ とチャレンジャー $\mathcal{B}$ 間の SIPA ゲームによって定義される。

セットアップ: チャレンジャー $\mathcal{B}$ は $(pk, sk) \leftarrow \mathsf{SSetup}(1^k)$ を実行する。また、空のリストを用意する。$\mathcal{B}$ は pk を攻撃者 $\mathcal{A}_1$ に入力する。その後、$\mathcal{A}_1$ は以下のクエリを適応的に行うことができる。

ユーザ生成クエリ UCreate: $\mathcal{A}_1$ が ID_i をクエリしてきたときには、$\mathcal{B}$ は $(pw_i, rk_i) \leftarrow \mathsf{Gen}(sk, ID_i)$ を実行し、リストに (ID_i, pw_i, rk_i) を追加する。$\mathcal{B}$ は $\mathcal{A}_1$ に何も返さない。また、$\mathcal{A}_1$ が以下のオラクルに ID をクエリするときには、その ID を事前に UCreate クエリしておかなければならない。

パスワード漏洩クエリ PassRev: $\mathcal{A}_1$ が ID_j をクエリしてきたときには、$\mathcal{B}$ はリストから pw_j を探して $\mathcal{A}_1$ に返す。

再発行鍵漏洩クエリ RecKeyRev: $\mathcal{A}_1$ が ID_ℓ をクエリしてきたときには、$\mathcal{B}$ はリストから rk_ℓ を探して $\mathcal{A}_1$ に返す。

再発行パスワード生成クエリ RecPassGen(RPG): $\mathcal{A}_1$ が ID_n をクエリしてきたときには、$\mathcal{B}$ は $(pw'_U, pw'_S) \leftarrow \mathsf{RePGen}(U(ID_m, ID_S, rk_m) \leftrightarrow S(ID_m, rk_m, pw_m, sk))$ を実行し、その実行手順中で通信路を流れる通信系列である $trans_{\mathsf{RPG}}$ を $\mathcal{A}_1$ に返す。この $trans_{\mathsf{RPG}}$ には再発行パスワード pw'_U, pw'_S は含まれない。その後、$\mathcal{B}$ はリストに $trans_{\mathsf{RPG}}$ を追加する。

認証クエリ Auth: $\mathcal{A}_1$ が ID_m をクエリしてきたときには、$\mathcal{B}$ は $\mathsf{Auth}(U(ID_m, ID_S, pw_m) \leftrightarrow S(ID_m, pw_m, sk))$ を実行し、その手順中で通信路を流れる取引履歴 $trans_{\mathsf{Auth}}$ と出力結果 $\top/\bot$ を $\mathcal{A}_1$ に返す。その後、$\mathcal{B}$ はリストに $trans_{\mathsf{Auth}}$ を追加する。

出力: $\mathcal{A}_1$ は (ID^*, st) を出力する。

$\mathcal{A}$ が勝利するためには $\mathcal{A}_1$ の出力する ID^* が

1. ID^* が UCreate クエリされている

2. $\mathcal{A}_1$ が ID^* に対する再発行鍵 rk^* を RecKeyRev クエリによって入手していない

3. $\mathcal{A}_1$ が ID^* に対する現在のパスワード pw^* を PassRev クエリによって入手していない

という 3 つの条件を満たしている必要がある。この条件を満たしている場合、$\mathcal{B}$ は st を $\mathcal{A}_2$ に入力する。その後、$\mathcal{A}_2$ と $\mathcal{B}$ は $\mathsf{Auth}(U(ID^*, ID_S, \cdot) \leftrightarrow S(ID^*, pw^*, sk))$ を実行する。この実行の途中で $\mathcal{A}_2$ は $\mathcal{A}_1$ と同じオラクルにクエリを発行することができる。しかし、パスワード漏洩オラクル、再発行鍵漏洩オラクル、再発行パスワード生成オラクルに ID^* をクエリすることはできない。最終的に Auth アルゴリズムの出力として $\mathcal{B}$ に $\top$ が出力されれば $\mathcal{A}$ の勝ちとなる。

定義 3.1 全ての多項式時間攻撃者 $\mathcal{A}$ に対し、$\mathsf{Adv}^{\mathsf{SIPA}}_{\mathcal{A}}(k) = \Pr[\mathcal{A}\ wins]$ が無視できるとき、パスワード再発行プロトコルは受動的成りすまし攻撃に対して安全であるという。

4 方式の構成と安全性証明

本章では、3 章で示されたモデルと安全性定義を満たすパスワード再発行プロトコルの構成、及びその安全性証明を示す。

4.1 ID ベース鍵共有とメッセージ認証コードを用いた一般的構成

本節では 2 章で示された ID ベース鍵共有とメッセージ認証コードを構成要素として用いたパスワード再発行プロトコルの一般的構成法を示す。ID ベース鍵共有における秘密鍵 sk_{ID} をパスワード再発行鍵 rk とし、その rk を用いて鍵共有を行うことで得られる鍵 K をパスワード pw と考える。パスワードを再発行する際には rk を用いて再度鍵共有を行うことで新たなパスワードを得ることができる。認証時には、ユーザはパスワードを用いて乱数のメッセージ認証コードを生成することで、正しいパスワードを保持していることの知識証明を行う。以上が方式の直感的な説明である。方式の詳細は図 1 の通り。

4.2 安全性証明

本節では図 1 の方式の安全性証明を示す。

$\mathsf{SSetup}(1^k)$:
$(mpk, msk) \leftarrow \mathsf{XSetup}(1^k)$
$sk_{ID_S} \leftarrow \mathsf{XExt}(msk, ID_S)$
$pk := (mpk, ID_S)$; $sk := (msk, sk_{ID_S})$
return (pk, sk).

$\mathsf{Gen}(sk, ID)$:
$sk := (msk, sk_{ID_S})$
$sk_{ID_U} \leftarrow \mathsf{XExt}(msk, ID_U)$
Server executes
$(K_U, K_S) \leftarrow \mathsf{XKE}(U(ID_S, sk_{ID_U}) \leftrightarrow S(ID_U, sk_{ID_S}))$
by itself.
$pw := K_U$; $rk := sk_{ID_U}$
return (pw, rk).

$\mathsf{RePGen}(U(ID_U, ID_S, rk) \leftrightarrow S(ID_U, rk, pw, sk))$:
$rk := sk_{ID_S}$
$(K'_U, K'_S) \leftarrow \mathsf{XKE}(U(ID_S, sk_{ID_U}) \leftrightarrow S(ID_U, sk_{ID_S}))$
$pw'_U := K'_U$; $pw'_S := K'_S$
return (pw'_U, pw'_S).

$\mathsf{Auth}(U(ID_U, ID_S, pw) \leftrightarrow S(ID_U, pw, sk))$:
$K := pw$
1. Server chooses random r and sends it to user.
2. User executes $tag \leftarrow \mathsf{MMac}(K, r)$ and sends tag to server.
3. Server executes $\mathsf{MVer}(K, r, tag)$.
   If the output is $\top$, return $\top$.
   else return $\perp$.

図 1: 提案するパスワード再発行プロトコルの一般的構成

定理 4.1 構成要素の *ID* ベース鍵共有方式が *2.2* 節で示した安全性定義を満たし、かつメッセージ認証コード方式が *EUF-CMA* 安全であるならば、図 *1* のパスワード再発行プロトコルは *SIPA* 安全である。

$\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ をパスワード再発行プロトコルの SIPA 攻撃者とする。q を $\mathcal{A}_1$ が発行する UCreate クエリの発行回数とし、q' を $\mathcal{A}_1$ が発行する RecPassGen クエリの発行回数とする。ここで q, q' は多項式である。以下のゲーム列を考える。

Game 0: 通常の SIPA ゲームとする。

Game 1: チャレンジャー $\mathcal{B}$ は、q 回の UCreate クエリのうち、何回目のクエリが $\mathcal{A}_1$ のチャレンジに使われるかを一様ランダムに予想する。これを i^* 回目とする。また、$\mathcal{B}$ は q' 回の RecPassGen クエリのうち、何回目の再発行パスワードが $\mathcal{A}_1$ のチャレンジに使われるかを一様ランダムに予想する。これを j^* 回目とする。ゲームを実行する中で i^*, j^* の推測が外れたと分かった時には、$\mathcal{B}$ は $\mathcal{A}$

がゲームに負けたと強制的に判定する。その他は Game 0 と同様とする。

Game 2. i^*, j^* に対応する部分のパスワードを ID ベース鍵共有方式を用いて生成された鍵から乱数に置き換える。その他は Game 1 と同様とする。

Game i において攻撃者 $\mathcal{A}$ が勝利する事象を W_i と書くことにする。$\mathsf{Adv}^{\mathsf{SIPA}}_{\mathcal{A}}(k) = \Pr[W_0]$ は定義より明らかなので、$\Pr[W_0] \leq qq' \Pr[W_1]$ であること、及び $qq' \Pr[W_1] \leq qq'(|\Pr[W_1] - \Pr[W_2]| + \Pr[W_2])$ の右辺の各項が無視できる値であることを示すことで $\mathsf{Adv}^{\mathsf{SIPA}}_{\mathcal{A}}(k)$ が無視できることを示したことになる。以下の 3 つの補題を証明する。

補題 4.1 $\Pr[W_0] \leq qq' \Pr[W_1]$ である。

補題 4.2 *ID* ベース鍵共有方式が *2.2* 節で示した安全性定義を満たすならば $|\Pr[W_1] - \Pr[W_2]|$ は無視できる。

補題 4.3 メッセージ認証コード方式が *EUF-CMA* 安全であるならば、$\Pr[W_2]$ は無視できる。

補題 4.1 の証明　チャレンジャー $\mathcal{B}$ の推測は少なくとも確率 $\frac{1}{qq'}$ で当たり、$\mathcal{B}$ の推測は $\mathcal{A}$ の行動とは全く独立であるため、$\mathcal{A}$ の動作に影響を与えない。よって、$\Pr[W_1] \geq \frac{1}{qq'} \Pr[W_0]$ が成り立つ。故に、$\Pr[W_0] \leq qq' \Pr[W_1]$ である。

補題 4.2 の証明　$\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ をパスワード再発行プロトコルの SIPA 攻撃者とし、$\mathcal{A}$ を用いて ID ベース鍵共有方式の攻撃者 $\mathcal{B}$ を構成する。$\mathcal{B}$ はチャレンジャーから mpk を受け取り、その mpk を $\mathcal{A}_1$ に入力する。その後、$\mathcal{B}$ は問題の埋め込みを行いつつ、$\mathcal{A}_1$ がオラクルに発行するクエリの返答をシミュレートする。

問題の埋め込み　$\mathcal{B}$ は、i^* 回目の UCreate クエリに用いられた ID（これを $ID^{\#}$ とする）に対する j^* 回目の RecPassGen クエリについては以下のように応答する。$\mathcal{B}$ は、Test オラクルに $ID^{\#}$ を問い合わせ、K_b を得る。その後、$\mathcal{B}$ は $\mathcal{A}_1$ に K_b を返す。その後、リストに $pw^{\#} := K_b$ を追加する。それ以外の場合に関しては以下のように応答する。

オラクルのシミュレート　$\mathcal{A}_1$ が UCreate オラクルに ID_i を問い合わせてきた場合には、$\mathcal{B}$ はその ID_i を Init オラクルに問い合わせる。

$\mathcal{A}_1$ が PassRev オラクルに ID_j を問い合わせてきた場合には、$\mathcal{B}$ はリストに pw_i があればそれを $\mathcal{A}_1$ に返す。リストに pw_i がなければ $\mathcal{B}$ は ID_j を XExt オラクルに問い合わせ、sk_{ID_j} を得る。その後、$\mathcal{B}$ は $\mathsf{XKE}(U(ID_S, sk_{ID_j}) \leftrightarrow S(ID_j, sk_{ID_S}))$ を一人で実行して pw_j を生成し、pw_j を $\mathcal{A}_1$ に返す。その後、$\mathcal{B}$ はリストに (ID_j, sk_{ID_j}, pw_j) を保存する。

$\mathcal{A}_1$ が RecKeyRev オラクルに ID_ℓ を問い合わせてきた場合には、$\mathcal{B}$ はリストに ID_ℓ があれば sk_{ID_ℓ} を探す。リストに ID_ℓ がなければ XExt オラクルに ID_ℓ を問い合わせ、sk_{ID_ℓ} を得た後、リストに $(ID_\ell, sk_{ID_\ell}, \cdot)$ を追加する。その後、$\mathcal{B}$ は $rk := sk_{ID_\ell}$ を $\mathcal{A}_1$ に返す。

$\mathcal{A}_1$ が RecPassGen オラクルに ID_m を問い合わせてきた場合には、$\mathcal{B}$ は ID_m を Trans オラクルに問い合わせ、$trans$ と $(K'_m, K'_S) = (pw'_m, pw'_S)$ を得る。その後、$\mathcal{B}$ は $trans_{\mathsf{RPG}} := trans$ を $\mathcal{A}_1$ に返す。その後、$\mathcal{B}$ はリストの pw_m を pw'_m に更新する。

$\mathcal{A}_1$ が Auth オラクルに ID_n を問い合わせてきた場合には、$\mathcal{B}$ はまず XExt オラクルに ID_n を問い合わせて sk_{ID_n} を得た後、$\mathcal{B}$ は $(pw_{n_U}, pw_{n_S} \leftarrow \mathsf{XKE}(U(ID_S, sk_{ID_n}) \leftrightarrow S(ID_n, sk_{ID_S}))$ を実行する。ただし、$ID_n = ID^{\#}$ であった場合には、$\mathcal{B}$ は XExt オラクルへの問い合わせ、及び鍵生成アルゴリズムの実行を行わず、リストから $pw^{\#} = K_b$ を探して以下の pw_{n_U} の代わりに $pw^{\#}$ を用いる。$\mathcal{B}$ はランダムな r_n を選び、$tag_n \leftarrow \mathsf{MMac}(pw_{n_U}, r_n)$ を実行して $trans_{\mathsf{Auth}} := (r_n, tag_n)$ と $\top$ を $\mathcal{A}_1$ に返す。

$\mathcal{A}_1$ は (ID^*, st) を出力して停止する。$\mathcal{A}$ が SIPA ゲームに勝利するという条件から、$\mathcal{A}_1$ が出力する ID^* は 3.2 節の条件を満たしている。

$\mathcal{A}_2$ への対応　その後、$\mathcal{B}$ は次のように動作する。$\mathcal{B}$ は st とランダムに選んだ r^* を $\mathcal{A}_2$ に入力する。$\mathcal{A}_2$ の発行するクエリに対し、$\mathcal{B}$ は先ほどと同じように応答することができる。ただし、ID^* に対する PassRev, RecKeyRev, RecPassGen の各クエリには $\perp$ を返す。$\mathcal{A}_2$ は tag^* を出力して停止するので、$\mathcal{B}$ は $\mathsf{MVer}(K_b, r^*, tag^*)$ を確認する。$\mathcal{B}$ は MVer の出力が $\top$ であれば $b' = 0$ を、$\perp$ であれば $b' = 1$ を出力して停止する。

以上が$\mathcal{B}$の記述である。$\mathcal{B}$のチャレンジビットbが0であれば$\mathcal{B}$はGame 0を、bが1であれば$\mathcal{B}$はGame 1を、$\mathcal{A}$に対してシミュレートできている。よって$|\Pr[W_1] - \Pr[W_2]| \leq \mathsf{Adv}^{\mathsf{IBKX}}_{\mathcal{B}}(k)$であるから、ID ベース鍵共有方式が安全であれば$|\Pr[W_1] - \Pr[W_2]|$は無視できることがわかる。

補題 4.3 の証明　$\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$をパスワード再発行プロトコルの SIPA 攻撃者とし、$\mathcal{A}$を用いてメッセージ認証コード方式の EUF-CMA 攻撃者 $\mathcal{B}$ を構成する。最初に $\mathcal{B}$ のチャレンジャーは $K^* \leftarrow \{0,1\}^k$ を生成する。まず $\mathcal{B}$ は $(mpk, msk) \leftarrow \mathsf{XSetup}(1^k)$ を実行し、mpk を $\mathcal{A}_1$ に入力する。その後 $\mathcal{B}$ は問題の埋め込みを行いつつ、$\mathcal{A}_1$ がオラクルに発行するクエリの返答をシミュレートする。

問題の埋め込み　$\mathcal{B}$ は補題 4.2 の証明と同様に i^*, j^* を一様ランダムに予想する。$\mathcal{A}_1$ が発行する i^* 回目の $\mathsf{UCreate}$ オラクルにクエリされた ID（これを $ID^{\#}$ とする）に対する j^* 回目の再発行パスワードの元での Auth クエリに対し、$\mathcal{B}$ は次のように応答する。$\mathcal{B}$ はまずランダムな $r^{\#}$ を選び、TagGen オラクルに $r^{\#}$ を問い合わせ、$tag^{\#}$ を得る。その後、$\mathcal{B}$ は $(r^{\#}, tag^{\#})$ と $\top$ を $\mathcal{A}_1$ に返す。それ以外の場合は以下のように応答する。

オラクルのシミュレート　$\mathcal{A}_1$ の $\mathsf{UCreate}, \mathsf{PassRev}, \mathsf{RecKeyRev}, \mathsf{RecPassGen}$ の各クエリに対し、$\mathcal{B}$ は msk を保持しているので全てのクエリに正当に答えることができる。

$\mathcal{A}_1$ は (ID^*, st) を出力して停止する。$\mathcal{A}$ が SIPA ゲームに勝利するという条件から、$\mathcal{A}_1$ が出力する ID^* は 3.2 節の条件を満たしている。

$\mathcal{A}_2$ への対応　その後、$\mathcal{B}$ は st とランダムに選んだ r^* を $\mathcal{A}_2$ に入力する。$\mathcal{A}_2$ の発行するクエリに対し、$\mathcal{B}$ は先ほどと同じように応答することができる。ただし、ID^* を $\mathsf{PassRev}, \mathsf{RecKeyRev}, \mathsf{RecPassGen}$ オラクルに問い合わせられた時には $\mathcal{B}$ は $\mathcal{A}_2$ に $\perp$ を返す。$\mathcal{A}_2$ は tag^* を出力して停止するので、$\mathcal{B}$ は (r^*, tag^*) のペアを出力して停止する。

以上が $\mathcal{B}$ の記述である。$\mathcal{A}$ の勝利条件から $\mathsf{MVer}(K^*, r^*, tag^*)$ の出力は $\top$ であることが保証されているため、$\Pr[W_2] \leq \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{B}}(k)$ である。よって、メッセージ認証コード方式が EUF-CMA 安全であれば $\Pr[W_2]$ は無視できることがわかる。

5　おわりに

本稿では ID とパスワードを用いたユーザ認証におけるパスワード再発行プロトコルのモデル、及び受動的攻撃に対する安全性を定義し、それらを満たす一般的構成法を提案した。今後の課題としては、能動的攻撃者や並行攻撃者といったより強力な攻撃者に対しても安全な方式の提案が挙げられる。また、今回の方式はパスワード再発行を無制限に行えるが、実運用上は再発行回数を制限したい場合もあると思われる。運用の仕方によって達成できることでもあるが、プロトコルとしてどのように対応するかを考察することも興味深い問題である。

謝辞　本研究の一部は JSPS 科研費 25280045 の助成によるものである。また、有益な意見をいただいた松浦研究室と新明るい暗号勉強会の皆様に感謝する。

参考文献

[1] Y. Dodis, J. Katz, S. Xu, M. Yung. Key-Insulated Public Key Cryptosystems. In *Proc. of EUROCRYPT2002*, LNCS2332, pages. 65–82, 2002.

[2] D. Fiore, R. Gennaro. Identity-Based Key Exchange Protocols without Pairings. In *Transactions on Computational Science*, Volume 10, pages. 42–77, 2010.

[3] N. Frykholm, A. Juels. Error-Tolerant Password Recovery. In *Proc. of CCS2001*, pages. 1–9, 2001.

[4] E. Okamoto. Key Distribution Systems Based on Identification Information. In *Proc. of CRYPTO1987*, LNCS293, pages. 194–202, 1988.

[5] R.W. Reeder, S. Schechter. When the Password Doesn't Work: Secondary Authentication for Websites. In *IEEE Security and Privacy*, Volume 9, No.2, pages. 43–49, 2011.

[6] R. Sakai, K. Ohgishi, M. Kasahara. Cryptosystems Based on Pairing. In *Proc. of SCIS2000*, 2000

[7] S. Schechter, R.W. .Reeder. 1+1=You: Measuring the comprehensibility of metaphors for configuring backup authentication. In *Proc. of SOUPS2009*, 2009.

[8] L. Standing, J. Conezio, R.N. Haber. Perception and memory for pictures: Single-trial learning of 2500 visual stimuli. In *Psychonomic Science*, Volume 19, Issue.2, pages. 73–74, 1970.

[9] 情報セキュリティ研究開発戦略 (改定版). 2014 年 7 月 10 日, `http://www.nisc.go.jp/active/kihon/pdf/kenkyu2014.pdf`.