証明可能安全なパスワード再発行プロトコル・改

大畑 幸矢 †‡　　松田 隆宏 ‡　　松浦 幹太 †

† 東京大学
〒 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
〒 135-0064 東京都江東区青海 2-4-7
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。著者らは CSS2014 において、このパスワード再発行プロトコルに証明可能安全性の考え方を導入し、モデルと安全性の定義、一般的構成の提案、及びその方式の安全性証明を行った。本稿ではその結果を拡張し、パスワード再発行プロトコルの新たなモデル、安全性定義、及び構成法について論じる。

Improved Provably Secure Password Recovery Protocol

Satsuya Ohata†‡　　Takahiro Matsuda‡　　Kanta Matsuura†

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

‡National Institute of Advanced Industrial Science and Technology (AIST)
2-4-7 Aomi, Koto-ku, Tokyo, 135-0064, Japan
t-matsuda@aist.go.jp

Abstract Many online services adopt a password-based user authentication system because of its usability. However, several problems have been pointed out on it, and one of the well-known problems is that a user forgets his/her password and cannot login the services. To solve this problem, most online services support a mechanism with which a user can recover a password. In CSS2014, the authors proposed a provably secure password recovery protocol. In this paper, we extend our previous result.

1　はじめに

1.1　背景

近年では社会の電子化が進み、多くのサービスがウェブ上で提供されるようになってきている。その際には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り、認証技術に関する研究は古くから活発に行われている。そのユーザ認証手法の中でも、本稿ではパスワードに基づくユーザ認証方式を取り上げる。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、ユーザが人間であるが故に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、これを克服するための研究（生体認証技術 [7] など）が現在でも多くなされている。しかし、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のが現状である。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。これは何らかの個人情報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。パスワード再発行プロトコルが脆弱であった場合、そのパスワードを用いて行われる元のプロトコルが（証明可能）安全であることの意味がなくなってしまう。このことからパスワード再発行プロトコルの安全性評価が重要な研究課題であることがわかるが、パスワード再発行プロトコルに関する既存の研究結果 [15, 9, 6, 11, 13, 14, 12, 8] は全て経験的な安全性評価（ヒューリスティクス）にとどまっていた。近年では [3] において、「秘密の質問」はアカウント復旧の仕組みとして不十分であることが指摘されており、より厳密な安全性評価を伴ったパスワード再発行プロトコルの研究を進めていく必要がある。

著者らは CSS2014 において、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を提案した [10]。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、ユーザがパスワードを忘れてしまうと、そのユーザは攻撃者と区別ができなくなってしまう。そこで、[10] においてはパスワードを再発行するための鍵（再発行鍵）を導入し、そのモデルの元で受動的攻撃者に対して安全な方式 (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) を提案した。しかし、[10] のモデル及び構成には、パスワードをユーザが任意に設定できないという問題があった [1]。

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再考する。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採用するのは [10] と同様であるが、今回のモデルではユーザがパスワードを自由に設定することが可能となっている。また、安全性定義として、パスワードを持たないユーザが認証に成功しないという安全性だけでなく、再発行鍵を持たない攻撃者はパスワードの再設定が出来ないという安全性も取り扱う。

まず本稿では、パスワード再発行プロトコルの新たなモデルの提案、及び安全性定義を行う。今回も [10] と同様に、受動的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擬似ランダム関数、認証暗号、公開鍵暗号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。構成要素の暗号（要素）技術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可能である。

1.3 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擬似ランダム関数、認証暗号、公開鍵暗号について、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。5 章では提案方式の拡張について述べる。

[1] ユーザがパスワードを選べないことがデメリットばかりとは言えない（脆弱なパスワードが選択される危険性が減る等）が、そもそもユーザが選べないパスワードはパスワードと呼べるのかどうかという問題がある。

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擬似ランダム関数、認証暗号、公開鍵暗号について振り返る。

2.1 表記法

S が集合ならば、$x \leftarrow S$ 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 は空の文字列を表す。PPT は確率的多項式時間 (Probabilistic Polynomial Time) の意味である。k は常にセキュリティパラメータを指すことにする。

2.2 擬似ランダム関数

以下の 2 つの条件を満足するとき、F は擬似ランダム関数であるという。

1. 鍵 $K \in \{0,1\}^k$ と文字列 x を入力として取り、$F(K, x)$ を出力する効率のよいアルゴリズムであること。
2. 攻撃者 $\mathcal{A}$ とチャレンジャー $\mathcal{B}$ 間で行われる次のような PRF ゲームを考える。まず、$\mathcal{B}$ はチャレンジビット $b \in \{0,1\}$ と鍵 $K \in \{0,1\}^k$ をランダムに選ぶ。$\mathcal{A}$ は適応的に x をクエリすることができる。このとき $\mathcal{B}$ は、$b = 0$ であれば $F(K, x)$ を、$b = 1$ であれば F の値域からランダムに選んだ値を $\mathcal{A}$ に返す。最終的に $\mathcal{A}$ は b の推測値 b' を出力する。$b = b'$ であれば $\mathcal{A}$ の勝ちであり、このゲームにおける $\mathcal{A}$ のアドバンテージ $\mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{A}}(k) = |\Pr[A\ wins] - 1/2|$ が全ての PPT 攻撃者に対して無視できること。

2.3 認証暗号

認証暗号 [2] は 2 つのアルゴリズム (AEEnc, AEDec) からなる。

AEEnc: 暗号化アルゴリズム AEEnc は、鍵 $Key \in \{0,1\}^k$ とメッセージ m を入力とし、暗号文 ct を出力する。これを $ct \leftarrow \mathsf{AEEnc}(Key, m)$ と書く。

AEDec: 復号アルゴリズム AEDec は、鍵 $Key \in \{0,1\}^k$ と暗号文 ct を入力とし、メッセージ m を出力する。これを $m \leftarrow \mathsf{AEDec}(Key, ct)$ と書く。

正当性 全ての鍵 $Key \in \{0,1\}^k$ とメッセージ $m \in \{0,1\}^*$ に対して、$\mathsf{AEDec}(Key, \mathsf{AEEnc}(Key, m)) = m$ となることが求められる。

安全性 認証暗号の安全性として、次の二つを考える。

まず、選択平文攻撃に対する識別不可能性 (Indistinguishability against Chosen Plaintext Attack, IND-CPA) は次のような攻撃者 $\mathcal{A}$ とチャレンジャー $\mathcal{B}$ 間で行われる INDCPA-AE ゲームによって定義される。まず、$\mathcal{B}$ はチャレンジビット $b \in \{0,1\}$ と鍵 $Key \in \{0,1\}^k$ を選ぶ。$\mathcal{A}$ は適応的に暗号化クエリとして長さの等しい二つのメッセージ (m_0, m_1) をクエリすることができ、$\mathcal{B}$ は $ct^* \leftarrow \mathsf{AEEnc}(Key, m_b)$ を計算して $\mathcal{A}$ に ct^* を返す。最終的に $\mathcal{A}$ は b の推測値 b' を出力して停止する。$b = b'$ であれば $\mathcal{A}$ の勝ちであり、このゲームにおける $\mathcal{A}$ のアドバンテージ $\mathsf{Adv}^{\mathsf{INDCPA\text{-}AE}}_{\mathcal{A}}(k) = |\Pr[A\ wins] - 1/2|$ が全ての PPT 攻撃者に対して無視できるとき、認証暗号は IND-CPA 安全であるという。

次に、暗号文の整合性（Integrity of Ciphertext, INT-CTXT）は次のような攻撃者 $\mathcal{A}$ とチャレンジャー $\mathcal{B}$ 間で行われる INTCTXT-AE ゲームによって定義される。まず、$\mathcal{B}$ は鍵 $Key \in \{0,1\}^k$ を選ぶ。$\mathcal{A}$ は適応的に暗号化クエリとして m をクエリすることができ、$\mathcal{B}$ は $ct^* \leftarrow \mathsf{AEEnc}(Key, m)$ を計算して $\mathcal{A}$ に ct^* を返す。最終的に $\mathcal{A}$ は暗号文 $\widehat{ct}$ を出力して停止する。$\mathcal{B}$ が暗号化クエリの応答結果として $\widehat{ct}$ を $\mathcal{A}$ に返した

ことがなく、かつ $\mathsf{AEDec}(Key, \widehat{ct}) \neq \perp$ であれば $\mathcal{A}$ の勝ちであり、このゲームにおける $\mathcal{A}$ のアドバンテージ $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{INTCTXT\text{-}AE}}(k) = \Pr[A\ wins]$ が全ての PPT 攻撃者に対して無視できるとき、認証暗号は INT-CTXT 安全であるという。

2.4 公開鍵暗号

公開鍵暗号 (Public Key Encryption, PKE) は 3 つのアルゴリズム $(\mathsf{PKG}, \mathsf{PEnc}, \mathsf{PDec})$ からなる。

PKG: 鍵生成アルゴリズム PKG は、セキュリティパラメータ 1^k を入力とし、公開鍵と復号鍵のペア (pk, dk) を出力する。これを $(pk, dk) \leftarrow \mathsf{PKG}(1^k)$ と書く。

PEnc: 暗号化アルゴリズム PEnc は、公開鍵 pk とメッセージ m を入力とし、暗号文 c を出力する。これを $c \leftarrow \mathsf{PEnc}(pk, m)$ と書く。

PDec: 復号アルゴリズム PDec は、復号鍵 dk と暗号文 c を入力とし、メッセージ m を出力する。これを $m \leftarrow \mathsf{PDec}(pk, c)$ と書く。

正当性 全ての $(pk, dk) \leftarrow \mathsf{PKG}(1^k)$ とメッセージ m に対して、$m = \mathsf{PDec}(dk, \mathsf{PEnc}(pk, m))$ となることが求められる。

選択暗号文攻撃に対する識別不可能性 (Indistinguishability against Chosen Ciphertext Attack, IND-CCA) は次のような攻撃者 $\mathcal{A}$ とチャレンジャー $\mathcal{B}$ 間で行われる INDCCA-PKE ゲームによって定義される。まず $\mathcal{B}$ はチャレンジビット $b \in \{0, 1\}$ を選び、$(pk, dk) \leftarrow \mathsf{PKG}(1^k)$ を実行して $\mathcal{A}$ に pk を入力する。その後、$\mathcal{A}$ はチャレンジクエリ（1 回のみ）と復号クエリを適応的に発行できる。チャレンジクエリとして、$\mathcal{A}$ は長さの等しい 2 つのメッセージ (m_0, m_1) を一度だけクエリすることができる。$\mathcal{B}$ は $c^* \leftarrow \mathsf{PEnc}(pk, m_b)$ を計算して $\mathcal{A}$ にチャレンジ暗号文 c^* を返す。復号クエリとして、$\mathcal{A}$ は暗号文 c をクエリすることができる。$\mathcal{B}$ は $m \leftarrow \mathsf{PDec}(dk, c)$ を計算して、$\mathcal{A}$ に m を返す。ただし、クエリされた c がチャレンジ暗号文であった場合には $\mathcal{B}$ は $\mathcal{A}$ に $\perp$ を返す。最終的に $\mathcal{A}$ は b の推測値 b' を出力して停止する。$b = b'$ であれば $\mathcal{A}$ の勝ちであり、このゲームにおける $\mathcal{A}$ のアドバンテージ $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{INDCCA\text{-}PKE}}(k) = |\Pr[A\ wins] - 1/2|$ が全ての PPT 攻撃者に対して無視できるとき、公開鍵暗号は IND-CCA 安全であるという。また、安全性証明を簡潔にするため、今回は複数回チャレンジが行える IND-CCA 安全性 (Multi-challenge IND-CCA, mIND-CCA) を考える。mIND-CCA 安全性は多項式程度の損失を除いては IND-CCA 安全性と等価になることが示されている [1]。詳細については割愛する。

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。

3.1 モデル

パスワード再発行プロトコルは 2 つのアルゴリズム $(\mathsf{SSetup}, \mathsf{RKG})$ と 2 つのやりとりを伴うアルゴリズム $(\mathsf{PRR}, \mathsf{Auth})$ 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、サーバの公開パラメータ pp と秘密鍵 sk を出力する。これを $(pp, sk) \leftarrow \mathsf{SSetup}(1^k)$ と書く。

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

PRR: パスワード（再）登録アルゴリズム PRR は、クライアント C_P が自身の ID とパスワード $pw \in \mathcal{PW}$、再発行鍵 rk を、サーバ S_P がクライアントの ID と再発行鍵 rk、秘密鍵 sk を入力し、C_P、S_P のローカルにそれぞれ ϕ、pw_s を出力する [2]。ここで $\mathcal{PW}$ はパスワード空間である。これを $(\phi, pw_s) \leftarrow$

[2] この pw_s はクライアントごとに定まるサーバ側の秘密情報という意味であり、pw と同じものでなくても構わない。

PRR$(C_P(ID, pw, rk) \leftrightarrow S_P(ID, rk, sk))$ と書く。[3]

Auth: 認証アルゴリズム Auth はクライアント C_A が自身の ID とパスワード pw を、サーバ S_A がクライアントの ID とサーバ側が保持している秘密情報 pw_s、秘密鍵 sk を入力し、C_A、S_A のローカルにそれぞれ ϕ、$\top/\perp$ を出力する。これを $(\phi, \top/\perp) \leftarrow$ Auth$(C_A(ID, pw) \leftrightarrow S_A(ID, pw_s, sk))$ と書く。

正当性 全ての $(pp.sk) \leftarrow$ SSetup(1^k)、$pw \in \mathcal{PW}$、$ID \in \{0,1\}^*$、$rk \leftarrow$ RKG(sk, ID)、$(\phi, pw_s) \leftarrow$ PRR$(C_P(ID, pw, rk) \leftrightarrow S_P(ID, rk, sk))$ に対して $(\phi, \top) \leftarrow$ Auth$(C_A(ID, pw) \leftrightarrow S_A(ID, pw_s, sk))$ とな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 RKG を用いてユーザごとに再発行鍵 rk を生成する。この rk は何らかの安全な方法でユーザに配送、保管されることを仮定しているが、この考え方は鍵隔離暗号 [5] に見られるものである [4]。パスワードの初期登録/再登録時には、ユーザは PRR をによって任意のパスワード pw をサーバに設定する。認証時には Auth によってユーザは自身を認証する。再発行鍵 rk はパスワードの初期登録/再登録時にのみ用いる。

3.2 安全性定義

以下の二種類の安全性を考える。

受動的なりすまし攻撃に対する安全性 (Impersonation under Passive Attack, Imp-PA) は、攻撃者 $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ とチャレンジャー $\mathcal{B}$ 間の Imp-PA 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 $(pp, sk) \leftarrow$ SSetup(1^k) を実行する。また、$(ID, pw, ps_s, rk, frag_p, frag_r)$ を保存するための空のリスト L を用意する。ここで、$frag_p$, $frag_r \in \{0,1\}$ である。$\mathcal{B}$ は pp 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。

クライアント生成クエリ CCreate: $\mathcal{A}_1$ が ID をクエリしてきたときには、$\mathcal{B}$ は次のように動作する。まず、$(ID, *, *, *, *, *)$ がリスト L に存在するとき、$\mathcal{B}$ は何もしない。そうでない時は $rk \leftarrow$ RKG(sk, ID) を実行し、$(ID, \perp, \perp, rk, 0, 1)$ をリスト L に追加する。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 CCreate クエリしておかなければならない。

再発行鍵漏洩クエリ RKR: $\mathcal{A}_1$ が ID をクエリしてきたときには、$\mathcal{B}$ はリスト L から $(ID, *, *, rk, *, *)$ を探し、rk を $\mathcal{A}_1$ に返す。その後、$\mathcal{B}$ はリストを $(ID, *, *, rk, *, 0)$ に更新する。

パスワード（再）登録クエリ PRR: $\mathcal{A}_1$ が (ID, pw') をクエリしてきたときには、$\mathcal{B}$ は次のように動作する（ここで、$pw' \in \mathcal{PW} \cup \{\phi\}$ である）。

$pw' \in \mathcal{PW}$ であれば、$\mathcal{B}$ はリスト L から $(ID, *, *, rk, *, *)$ を探し、$(\phi, pw'_s) \leftarrow$ PRR$(C_P(ID, pw', rk) \leftrightarrow S_P(ID, rk, sk))$ を実行してその通信履歴 $trans_{\mathsf{PRR}}$ を $\mathcal{A}_1$ に返す。その後、$\mathcal{B}$ はリスト L を $(ID, pw', pw'_s, rk*, 0, *)$ に更新する。

$(pw' = \phi)$ であれば、$\mathcal{B}$ はまずランダムなパスワード $pw' \in \mathcal{PW}$ を選ぶ。次に、$\mathcal{B}$ はリスト L から $(ID, *, *, rk, *, *)$ を探し、$(\phi, pw'_s) \leftarrow$ PRR$(C_P(ID, pw', rk) \leftrightarrow S_P(ID, rk, sk))$ を実行してその通信履歴 $trans_{\mathsf{PRR}}$ を $\mathcal{A}_1$ に返す。その後、$\mathcal{B}$ はリスト L を $(ID, pw', pw'_s, rk*, 1, *)$ に更新する。

認証クエリ Auth: $\mathcal{A}_1$ が (ID, pw') をクエリしてきたときには、$\mathcal{B}$ は次のように動作する（ここで、$pw' \in \mathcal{PW} \cup \{\phi\}$ である）。

[3] 提案方式はサーバ S_P 側の入力として秘密鍵を用いる必要がないが、一般には sk を用いることも許されるべきであり、拡張方式を考えると PRR において sk が必要になる場合もある。これは Auth に関しても同様である。

[4] このモデルは理論的なものであると思われがちであるが、このモデルで動作していると考えられる実用的なセキュリティ技術（セキュリティトークン等）も存在する。

$pw' \in \mathcal{PW}$ であれば、$\mathcal{B}$ はリスト L から $(ID, *, pw_s, rk, *, *)$ を探し、$(\phi, \top/\bot) \leftarrow \mathsf{Auth}(C_A(ID, pw') \leftrightarrow S_A(ID, pw_s, sk))$ を実行してその通信履歴 $trans_{\mathsf{Auth}}$ と S_P に出力された $\top/\bot$ を $\mathcal{A}_1$ に返す。

$pw' = \phi$ であれば、$\mathcal{B}$ はリスト L から $(ID, pw, pw_s, rk, *, *)$ を探し、$(\phi, \top/\bot) \leftarrow \mathsf{Auth}(C_A(ID, pw) \leftrightarrow S_A(ID, pw_s, sk))$ を実行してその通信履歴 $trans_{\mathsf{Auth}}$ と S_P に出力された $\top/\bot$ を $\mathcal{A}_1$ に返す。

出力: $\mathcal{A}_1$ は (ID^*, st) を出力して停止する。$\mathcal{A}$ が Imp-PA ゲームに勝利するためには、$(ID^*, pw^*, pw_s^*, rk^*, frag_p^*, frag_r^*)$ がリスト L に存在し、かつ $frag_p^* = frag_r^* = 1$ でなければならない。これらの条件を満たしていない場合には、$\mathcal{B}$ は $\mathcal{A}$ が Imp-PA ゲームに負けたと判断する。そうでなければ、$\mathcal{B}$ は st を $\mathcal{A}_2$ に入力する。$\mathcal{B}$ は $\mathcal{A}_2$ と $\mathsf{Auth}(\mathcal{A}_2(st) \leftrightarrow S_A(ID^*, pw_s^*, sk))$ を実行する。この実行中に、$\mathcal{A}_2$ は適応的に $\mathcal{A}_1$ と同じクエリを発行することができる。ただし、ID^* を RKR 及び PRR にクエリすることはできない。最終的に Auth の実行結果として S_A に $\top$ が発行されれば $\mathcal{A}$ の勝ちとなる。

定義 1 全ての多項式時間攻撃者 $\mathcal{A}$ に対して $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Imp\text{-}PA}}(k) = \Pr[\mathcal{A}\ wins]$ 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 [5]。

次に、受動的不正登録攻撃に対する安全性 (Illegal Registration under Passive Atttack, IR-PA) を考える。この安全性は攻撃者 $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ とチャレンジャー $\mathcal{B}$ 間の IR-PA ゲームによって定義される。セットアップ、$\mathcal{A}_1$ のクエリとそれに対する $\mathcal{B}$ の動作、最終的な $\mathcal{A}_1$ の出力 (ID^*, st) は Imp-PA ゲームと同様である。$\mathcal{A}$ がこの IR-PA ゲームに勝利するためには、$(ID^*, pw^*, pw_s^*, rk^*, frag_p^*, frag_r^*)$ がリスト L に存在し、かつ $frag_r^* = 1$ でなければならない。これらの条件を満たしていない場合には、$\mathcal{B}$ は $\mathcal{A}$ が Imp-PA ゲームに負けたと判断する。そうでなければ、$\mathcal{B}$ は st を $\mathcal{A}_2$ に入力する。$\mathcal{B}$ は $\mathcal{A}_2$ と $\mathsf{PRR}(\mathcal{A}_2(st) \leftrightarrow S_A(ID^*, rk^*, sk))$ を実行する。この実行中に、$\mathcal{A}_2$ は適応的に $\mathcal{A}_1$ と同じクエリを発行することができる。ただし、ID^* を RKR 及び PRR にクエリすることはできない。最終的に PRR の実行結果として S_A に $\bot$ と異なるものが発行されれば $\mathcal{A}$ の勝ちとなる。

定義 2 全ての多項式時間攻撃者 $\mathcal{A}$ に対して $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{IR\text{-}PA}}(k) = \Pr[\mathcal{A}\ wins]$ 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不正登録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のスケッチを示す。

4.1 擬似ランダム関数、認証暗号、公開鍵暗号を用いた一般的構成

本節では擬似ランダム関数、認証暗号、公開鍵暗号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が、その前に方式の直感的な説明を与える。ユーザが任意のパスワードを選び、それをどのようにしてサーバに（再）登録するかが問題となるが、今回は単純に、クライアントが設定したいパスワードを暗号化してサーバに送るという方法を考える。（再）登録時には再発行鍵を用いてよいので、この暗号化の際には再発行鍵を暗号化用の鍵として用いればよい。再発行鍵として擬似ランダム関数の出力を用いることで、暗号化用の鍵がランダムであることが保証される。しかし、暗号化の際に IND-CPA 安全性を持つ共通鍵暗号を用いると不正登録攻撃に対する安全性を証明できないので、INT-CTXT 安全性を持つ認証暗号を用いてパスワードを暗号化し、送信する。認証の際にはクライアントが通常の公開鍵暗号を用いてパスワードを暗号化、送信し、サーバ

[5] 記述の簡便さのため、パスワード空間のサイズ $|\mathcal{PW}|$ が超多項式的に大きいことを仮定している。それを避けたければ、「無視できる」ではなく、(Auth クエリ回数)$/|\mathcal{PW}| + negl.$ とすればよい。

$\mathsf{SSetup}(1^k)$: $(pk, dk) \leftarrow \mathsf{PKG}(1^k)$ ランダムな鍵 $K \in \{0,1\}^k$ を選ぶ。 $pp := pk$; $sk := (K, dk)$ (pp, sk) を出力する。
$\mathsf{RKG}(sk, ID)$: $(K, dk) \leftarrow sk$ $rk := F(K, ID)$ rk を出力する。
$(\phi, pw_s) \leftarrow \mathsf{PRR}(C_P(ID, pw, rk) \leftrightarrow S_P(ID, rk, sk))$: $Key \leftarrow rk$ 1. S_P は乱数 $r \in \{0,1\}^k$ を選び、C_P に送る。 2. C_P は $ct \leftarrow \mathsf{AEEnc}(Key, r\|pw)$ を実行し、 ct を S_P に送る。 3. S_P は $r'\|pw_s \leftarrow \mathsf{AEDec}(Key, ct)$ を実行する。 4. もし $r' = r$ が成り立てば S_P は pw_s を出力する。 そうでなければ S_P は $\perp$ を出力する。
$(\phi, \top/\perp) \leftarrow \mathsf{Auth}(C_A(ID, pw) \leftrightarrow S_A(ID, pw_s, sk))$: 1. S_A は乱数 $r \in \{0,1\}^k$ を選び、C_A に送る。 2. C_A は $c \leftarrow \mathsf{PEnc}(pk, r\|pw)$ を実行し、 c を S_A に送る。 3. S_A は $r'\|pw' \leftarrow \mathsf{PDec}(dk, c)$ を実行する。 4. もし $r' = r$ と $pw' = pw_s$ が成り立てば S_A は $\top$ を 出力し、そうでなければ $\perp$ を出力する。

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

側で登録されているパスワードと等しいかどうかを確認すればよい。パスワード（再）登録・認証ともに、サーバ側から送られてきた乱数とパスワードを連結して暗号化するという手順になっているが、これは再送攻撃を防ぐためである。また、認証の際には乱数のメッセージ認証コードを生成して送信すればよいとも考えられるが、公開鍵暗号を用いているのは辞書攻撃を防ぐためである。以上が方式の直感的な説明である。方式は図 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のスケッチを示す。

定理 1 *F が擬似ランダム関数であり、AE が IND-CPA 安全であり、PKE が IND-CCA 安全*[6] *であるならば、図 1 のパスワード再発行プロトコルは Imp-PA 安全である。*

[6]厳密に言えば 1-bounded CCA 安全性で十分であり、理論的には CPA 安全な公開鍵暗号から (1-)bounded CCA 安全な公開鍵暗号が構成できることが知られている [4]。

安全性証明のスケッチ Imp-PA 安全性は直感的には pw を持たないクライアントが認証に合格できないことを意味する安全性なので、その証明においては、RKR, PRR, Auth の各クエリの返答から pw(と K) の情報を消していけばよい。ゲーム変換を用いて証明を行うが、RKR クエリの返答から情報を消す際には擬似ランダム関数の安全性を、PRR クエリの返答から情報を消す際には認証暗号の INDCPA-AE 安全性をそれぞれ用いる。Auth クエリの返答から情報を消す際には公開鍵暗号の安全性に帰着させるが、クエリの返答に関しては INDCPA-PKE 安全性でよい。しかし、最終的な攻撃者の出力（＝暗号文）が勝利条件を満たしているかどうかをチャレンジャーが確認するために 1 度だけ復号クエリが必要となり、そのため帰着させる安全性は INDCCA-PKE 安全性となる。比較的テクニカルな部分としては、乱数の衝突を防ぐためのゲーム変換が必要になること、及び Auth クエリの返答から情報を消すゲームに変換する前にチャレンジで用いられる ID が何番目に CCreate クエリされたものなのかを推測する必要があることが挙げられる。この推測によって帰着が $1/q_c$（ここで q_c は攻撃者によって発行される CCreate クエリの回数）だけ損失する。

定理 2 *F が擬似ランダム関数であり、AE が INT-CTXT 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 IR-PA 安全である。*

安全性証明のスケッチ IR-PA 安全性は、再発行鍵を持たないクライアントはパスワードの（再）発行ができないということを意味する安全性なので、その証明においては RKR クエリの返答から情報を消せばよい。このためには、Imp-PA 安全性の証明と同様に擬似ランダム関数の安全性を用いてゲーム変換を行う。再発行鍵から情報を消した後は、認証暗号の INT-CTXT 安全性への帰着が可能となる。

5 拡張方式

4 章で提案したプロトコルはサーバにクライアントのパスワードそれ自体を保存しているが、

$\mathsf{SSetup}(1^k)$: $(pk, dk) \leftarrow \mathsf{PKG}(1^k)$ ランダムな鍵 $K \in \{0,1\}^k$ を選ぶ。 $pp := pk$; $sk := (K, dk)$ (pp, sk) を出力する。
$\mathsf{RKG}(sk, ID)$: $(K, dk) \leftarrow sk$ $rk := F(K, 0\|\|ID)$ rk を出力する。
$(\phi, pw_s) \leftarrow \mathsf{PRR}(C_P(ID, pw, rk) \leftrightarrow S_P(ID, rk, sk))$: $Key \leftarrow rk$ 1. S_P は乱数 $r \in \{0,1\}^k$ を選び、C_P に送る。 2. C_P は $ct \leftarrow \mathsf{AEEnc}(Key, r\|pw)$ を実行し、 ct を S_P に送る。 3. S_P は $r'\|pw' \leftarrow \mathsf{AEDec}(Key, ct)$ を実行する。 4. S_P は $pw_s = F(K, 1\|\|pw')$ 5. もし $r' = r$ が成り立てば S_P は pw_s を出力する。 そうでなければ S_P は $\perp$ を出力する。
$(\phi, \top/\perp) \leftarrow \mathsf{Auth}(C_A(ID, pw) \leftrightarrow S_A(ID, pw_s, sk))$: 1. S_A は乱数 $r \in \{0,1\}^k$ を選び、C_A に送る。 2. C_A は $c \leftarrow \mathsf{PEnc}(pk, r\|pw)$ を実行し、 c を S_A に送る。 3. S_A は $r'\|pw' \leftarrow \mathsf{PDec}(dk, c)$ を実行する。 4. もし $r' = r$ と $pw_s = F(K, 1\|\|pw')$ が成り立てば S_A は $\top$ を出力し、そうでなければ $\perp$ を出力する。

図 2: パスワード再発行プロトコルの拡張方式

これは一般的には好ましくないこととされている。今回提案している方式においてはそれを避けることも可能であり、具体的にはサーバの秘密鍵を用いてパスワードを変形させて保存する方式（簡易的なソルティング）に拡張可能である。この方式は、PRR や Auth アルゴリズムにおいてサーバ側は秘密鍵 sk を用いてもよいことを利用しており、定義域分割技法（擬似ランダム関数の入力の先頭ビットを固定し、同じ鍵を異なる目的に使う技法）を用いることでサーバ側に新たな鍵を追加しなくともよい方式になっている。安全性証明も 4 節の方式とほぼ同様に可能であるが、本稿では割愛する。方式は図 2 の通り。

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、本研究に関して有益な意見をいただいた松浦研究室、及び新明るい暗号勉強会の皆様に感謝する。

参考文献

[1] M. Bellare, A. Boldyreva, and S. Micali. Public Key Encryption in a Multi-user Setting: Security Proofs and Improvements. In *Proc. of EUROCRYPT 2000*, LNCS 1807, pp. 259–274, 2000.

[2] M. Bellare, C. Namprempre. Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm. In *J. Cryptology*, Volume 21, No.4, pages. 469–491, 2008.

[3] J Bonneau, E. Bursztein, R. Jackson, M. Williamson. Secrets, lies, and account recovery: Lessons from the use of personal knowledge questions at Google. In *Proc. of WWW2015*, pp.141-150, 2015.

[4] R. Cramer, G. Hanaoka, D. Hofheinz, H. Imai, E. Kiltz, R. Pass, A. Shelat, V. Vaikuntanathan. Bounded CCA2-Secure Encryption. In *Proc. of ASIACRYPT2007*, LNCS4833, pages. 502–518, 2007.

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

[6] M. Jakobsson, E. Stolterman, S. Wetzel, L. Yang. Love and Authentication. In *Proc. of CHI2008*, pages. 197–200, 2008.

[7] A.K. Jain, A. Ross, S. Prabhakar. An Introduction to Biometric Recognition. In *IEEE Transactions on Circuits and Systems for Video Technology*, Volume 14, No.1, pages. 4–20, 2004.

[8] A. Javed, D. Bletgen, F. Kohlar, M. Dürmuth, J. Schwenk. Secure Fallback Authentication and the Trusted Friend Attack. In *Proc. of ICDCSW2014*, pages. 22–28, 2014.

[9] M. Just, D. Aspinall. Personal Choice and Challenge Questions: A Security and Usability Assessment. In *Proc. of SOUPS2008*, 2008.

[10] 大畑幸矢, 松田隆宏, 松浦幹太. 証明可能安全なパスワード再発行プロトコルについて. コンピュータセキュリティシンポジウム 2014, 3E2-2, 2014.

[11] A. Rabkin. Personal Knowledge Questions for Fallback Authentication: Security Questions in the Era of Facebook. In *Proc. of SOUPS2008*, 2008.

[12] 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.

[13] S.E. Schechter, A.J.B. Brush, S. Egelman. It's No Secret. Measuring the Security and Reliability of Authentication via "Secret" Questions. In *Proc. of IEEE Symposium on Security and Privacy*, pages. 375–390, 2009.

[14] S. Schechter, R.W. Reeder. 1+1=You: Measuring the Comprehensibility of Metaphors for Configuring Backup Authentication. In *Proc. of SOUPS2009*, 2009.

[15] M. Zviran, W.J. Haga. User Authentication by Cognitive Passwords: An Empirical Assessment. In *Proc. of JCIT90*, pages. 137–144, 1990.