令和 6 年度

修士学位論文

パスワード管理システムに関する研究

A Study on Password Management Systems

1275095　　青木 友志

指導教員　　清水 明宏

2025 年 2 月 28 日

高知工科大学大学院 工学研究科 基盤工学専攻

情報学コース

要 旨

パスワード管理システムに関する研究

青木 友志

インターネットの普及に伴い Web サービスはその種類も数も増加している．それに比例してユーザ個人が管理する必要のある ID とパスワードの組も増加している．しかし，記憶しておける数には限界がある点や，同じものを使いまわす場合セキュリティが低くなってしまう．これらの対策として，パスワード管理システムがある．しかし，その多くは固定パスワードによる認証が採用され，これは固定パスワードが漏洩すると保存している ID とパスワードの組も漏洩するため，不正利用される危険性が挙げられる．また，パスワード管理システムの従来の方式では公開鍵系の暗号方式を用いた鍵配送の方式があるが，これは鍵交換に時間がかかるという課題がある．これらの課題を解決する方法としてワンタイムパスワード認証方式に基づいた鍵配送の方式が提案されているが，ワンタイムパスワードは通信ごとに認証情報が変化し鍵の更新が困難であるため，複数端末での利用が困難である．この課題の解決には渡邊の方式，高橋の方式などが提案されているが，どちらの方式も認証情報の管理コストが高くなり効率的にワンタイムパスワードを利用できていない．そこで本研究では，既存方式の問題点を解決した複数端末認証が可能なワンタイムパスワードを効率よく利用する方式を提案する．既存方式と比較して，提案方式は認証情報の管理コストおよび一方向性関数の適用回数の面で有用性を示し，ワンタイムパスワードを効率よく利用できていることを示した．

キーワード　パスワード管理システム，複数端末認証，ワンタイムパスワード認証方式，SAS-X，SAS-L

Abstract

A Study on Password Management Systems

Yuji Aoki

With the spread of the Internet, the number and variety of Web services are increasing. In proportion to this, the number of ID and password pairs that each user needs to manage is also increasing. However, there is a limit to the number of IDs and passwords that can be stored in memory, and security is reduced when the same IDs and passwords are used repeatedly. Password management systems are available as a countermeasure for these problems. However, most of them use fixed passwords for authentication, and if a fixed password is compromised, the stored ID and password pairs will also be compromised, posing the risk of unauthorized use. Another conventional method for password management systems is key distribution using public-key cryptography, but this method has the problem that key exchange takes time. However, the one-time password authentication method is difficult to use in multiple terminals because the authentication information changes with each communication and it is difficult to update the key. However, it is difficult to use one-time passwords at multiple terminals because the authentication information changes with each communication and it is difficult to update the key. In this study, we propose a method for efficient use of one-time passwords that solves the problems of existing methods and enables multi-terminal authentication. Compared with existing methods, the proposed method is more effective in terms of the management cost of authentication information and the number of times the one-way function is applied, and we show that the one-time password can be used efficiently.

key words Password Management System, Multi Device Authentication, One-time password authentication method, SAS-X, SAS-L

目次

図目次

表目次

第 1 章

はじめに

1.1　背景

インターネットの普及に伴い Web サービスはその種類も数も大きく増加しており，近年では様々なサービスを受けることが可能である．総務省によるインターネットの利用率の調査では，2022 年では 13 歳から 59 歳までの各階層で 9 割を，6 歳から 12 歳および 60 歳から 69 歳の階層も 8 割を超えており，2023 年では 60 歳から 69 歳までの階層でも 9 割を超え，6 歳から 12 歳の階層でも 9 割に迫っている [1]．総務省によるインターネットの利用率の調査結果を図 1.1 に示す．

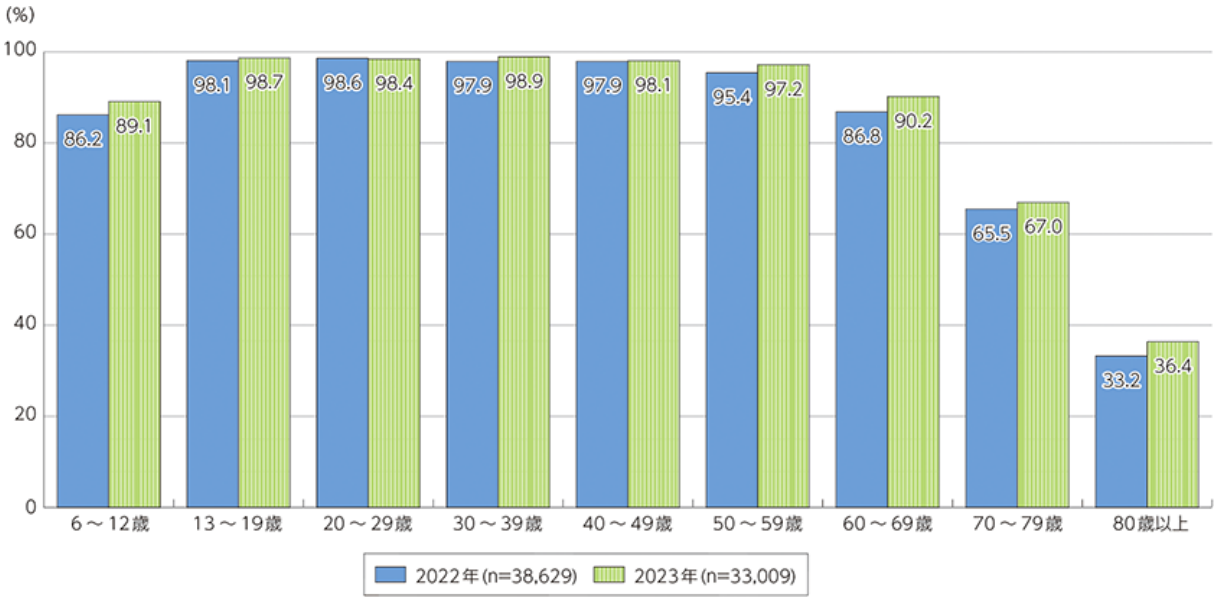


図 1.1　総務省による年齢別インターネット利用率の調査結果

また，総務省によるインターネット利用の目的・用途の調査では最も高いのは「SNS（無

1.1　背景

料通話機能を含む）の利用」で 80.8%であり,「電子メールの送受信」76.0%,「情報検索」74.7%,「商品・サービスの購入・取引」59.9%と続く [2]．総務省によるインターネットの利用の目的・用途の調査結果を図 1.2 に示す.

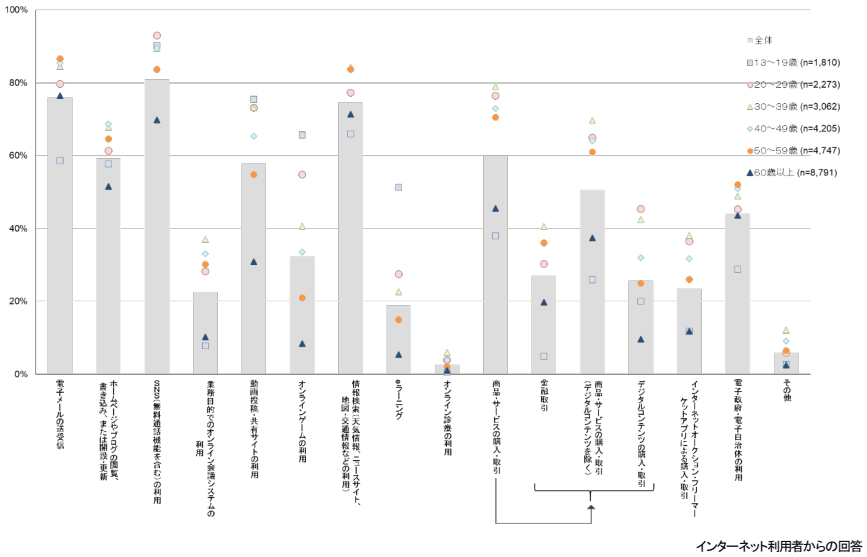


図 1.2　総務省によるインターネット利用の目的・用途の調査結果

上記の調査結果より，日常生活でインターネットのサービスを利用する機会は多いといえる．一般的なインターネットのサービスは利用者ごとにアカウントを作成し，固定の ID とパスワードの組で認証が行われる．すなわち，ユーザはサービスごとに ID とパスワードの組を入力する必要があり，利用するサービスの数が増加するとユーザ個人が管理する ID とパスワードの組数も増加することになる．しかし，トレンドマイクロ株式会社によるパスワード数の調査によると 83.8%もの人がパスワードを使いまわしていることが判明しており，セキュリティの面で非常に危険といえる [3].

また，同社によると，パスワードを使いまわす理由としては「異なるパスワードを設定すると忘れてしまう」が 72.8%,「異なるパスワードを考えるのが面倒」が 48.6%を占めており，パスワードを忘れてしまうという理由が最多数であった [3]．トレンドマイクロ株式会社による利用サービス数とパスワード数の調査結果をおよびパスワードを使いまわす理由を図 1.3 および図 1.4 に示す.

1.1 背景

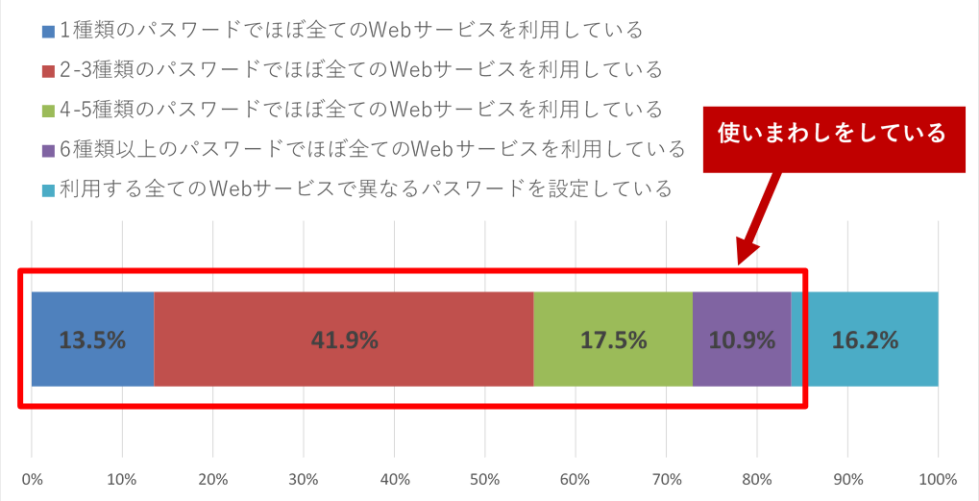


図 1.3　トレンドマイクロ株式会社による利用サービス数とパスワード数の調査結果

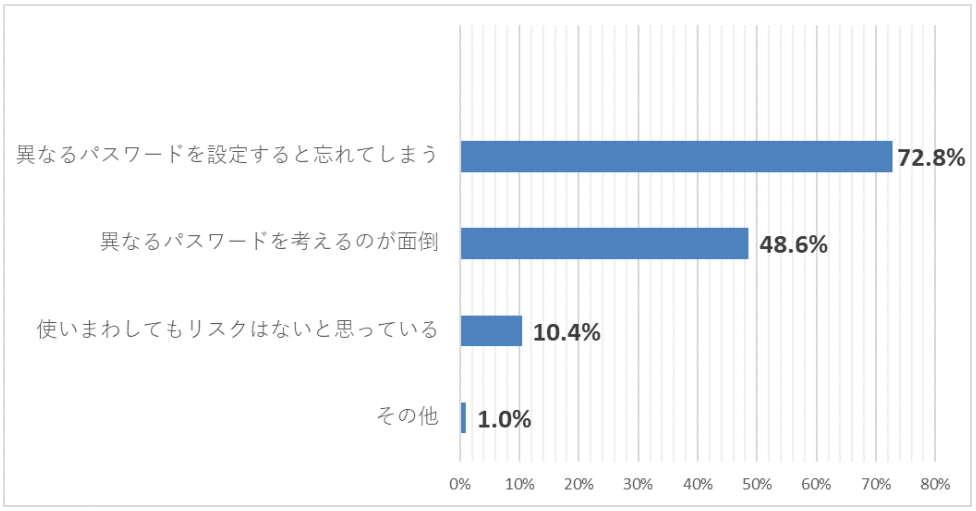


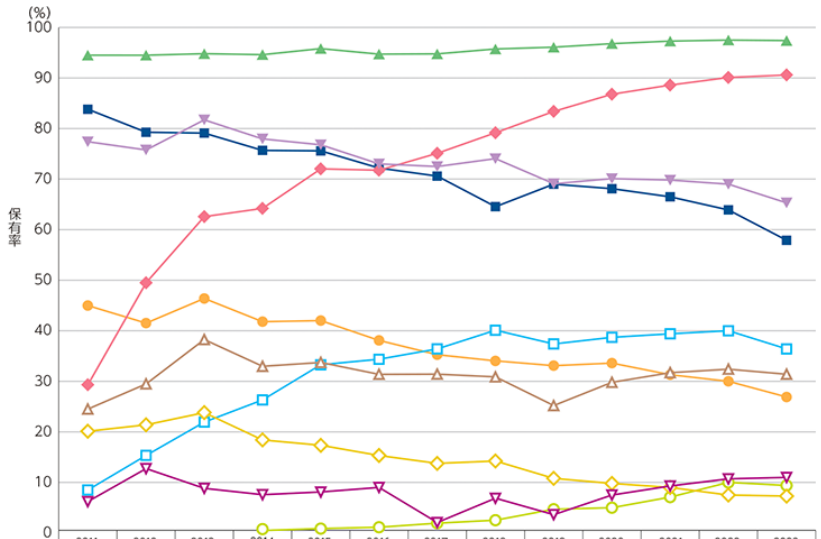
図 1.4　トレンドマイクロ株式会社によるパスワードを使いまわす理由の調査結果

上記の危険性の対策として，パスワード管理システムと呼ばれるものがある．パスワード管理システムとは，ユーザが利用しているサービスの ID とパスワードを一元管理できるシステムであり，このシステムを用いることでユーザは多くの ID とパスワードの組を覚えておく必要がなくなる．しかし，既存の多くのパスワード管理システムには固定パスワードによる認証が採用されており，この認証方式では使用している固定パスワードが漏洩すると保存している ID とパスワードの組も漏洩するため，不正利用される危険性が挙げられる．

また，このようなパスワード管理システムではパスワードの登録や参照を行うごとに安全

な鍵共有のための鍵配送を行う必要がある．従来の方式では Diffie-Hellman 法，RSA 暗号，楕円曲線暗号などでの方式が用いられるが，これらの方式は公開鍵系の暗号方式であり計算コストが高く，鍵交換に時間がかかるという課題がある．

このセキュリティや鍵配送の課題を解決する方法として，ワンタイムパスワード認証方式に基づいて認証情報を鍵生成の要素として用いる方式が提案されている．しかし，ワンタイムパスワードは通信のたびに認証情報が変化するため鍵の更新が困難であり，複数端末での利用が困難という問題点が挙げられる．総務省の調査では 2023 年度の情報通信機器の保有率は「スマートフォン」が 90.6%,「パソコン」が 65.3%，タブレット型端末が 36.4%となっている [4]．このことから，近年では複数端末から同一のサービスを利用するケースも増えているため，複数端末から利用可能であることは必須であるといえる．総務省による情報通信機器・端末の保有率の推移を図 1.5 に示す．



	2011 (n=16,530)	2012 (n=20,418)	2013 (n=15,599)	2014 (n=16,529)	2015 (n=14,765)	2016 (n=17,040)	2017 (n=16,117)	2018 (n=16,255)	2019 (n=15,410)	2020 (n=17,345)	2021 (n=17,365)	2022 (n=15,951)	2023 (n=14,009)
固定電話	83.8	79.3	79.1	75.7	75.6	72.2	70.6	64.5	69.0	68.1	66.5	63.9	57.9
FAX	45.0	41.5	46.4	41.8	42.0	38.1	35.3	34.0	33.1	33.6	31.3	30.0	26.9
モバイル端末全体	94.5	94.5	94.8	94.6	95.8	94.7	94.8	95.7	96.1	96.8	97.3	97.5	97.4
スマートフォン	29.3	49.5	62.6	64.2	72.0	71.8	75.1	79.2	83.4	86.8	88.6	90.1	90.6
パソコン	77.4	75.8	81.7	78.0	76.8	73.0	72.5	74.0	69.1	70.1	69.8	69.0	65.3
タブレット型端末	8.5	15.3	21.9	26.3	33.3	34.4	36.4	40.1	37.4	38.7	39.4	40.0	36.4
ウェアラブル端末				0.5	0.9	1.1	1.9	2.5	4.7	5.0	7.1	10.0	9.4
インターネットに接続できる家庭用テレビゲーム機	24.5	29.5	38.3	33.0	33.7	31.4	31.4	30.9	25.2	29.8	31.7	32.4	31.4
インターネットに接続できる携帯型音楽プレイヤー	20.1	21.4	23.8	18.4	17.3	15.3	13.8	14.2	10.8	9.8	9.0	7.5	7.3
その他インターネットに接続できる家電（スマート家電）等	6.2	12.7	8.8	7.6	8.1	9.0	2.1	6.9	3.6	7.5	9.3	10.7	11.0

図 1.5　総務省による情報通信機器・端末の保有率の推移

こういった問題点の対策として，ワンタイムパスワード認証方式の 1 種である SAS-X(Simple And Secure password authentication protocol，extra-secure version) を用いて複数端末認証を実現した渡邊の方式 [5] や高橋の方式 [6] がある．しかし，これらの方式はユーザの端末ごとに通信が必要であり固定パスワードで認証する方式と比べて認証情報の管理コストが高く，ワンタイムパスワードを効率的に利用できていないといえる．

そこで，本研究ではワンタイムパスワード認証方式の 1 種である SAS-L を用いて複数端末認証が可能であり，ワンタイムパスワードを効率よく利用できるパスワード管理システムを提案する．

1.2　本論文の構成

本論文は全 7 章で構成する．

第 2 章では，ワンタイムパスワード認証方式の 1 種である SAS について述べる．

第 3 章では，既存方式の手法およびその問題点について述べる．

第 4 章では，既存方式の問題点を解決した提案方式について述べる．

第 5 章では，評価について述べる．

第 6 章では，考察について述べる．

最後に本研究の成果をまとめ，今後の課題を述べる．

第 2 章

ワンタイムパスワード認証方式 SAS

本章では，ワンタイムパスワード認証方式の 1 種である SAS について述べる．

2.1 認証方式

認証方式は大きく分けて固定パスワード認証方式とワンタイムパスワード認証方式に分けられる．固定パスワード認証方式は，ユーザが認証時に入力した固定の値に一方向性関数を適用したものをサーバへ送信し認証を行う方式である．なお，ここで述べる一方向性関数とは，ある値 x に一方向性関数 H を適用すると全く異なる値 $H(x)$ になる関数であり，一般的に x から $H(x)$ を求めることは容易であるが，$H(x)$ から x を求めることは計算量的に困難であるという特徴を持つ．しかし，通信している情報は常に固定であるため通信を盗聴された場合，ユーザとサーバのみが知る情報を容易に取得することができてしまい，この情報を再利用されるとなりすましをされる危険性がある．

ワンタイムパスワード認証方式は，ユーザの認証に用いるパスワードを使い捨てにすることで，各認証時にパスワードが異なる方式である．ワンタイムパスワード認証方式は，通信を盗聴された場合でも次回認証時には異なるパスワードになっており，再利用される危険性がないため，固定パスワード認証方式の問題点であるなりすましを解決することができている．ワンタイムパスワード認証方式の種類には，ローリングコード方式，チャレンジ&レスポンス方式，SAS(Simple And Secure password authentication protocol) などがある．

2.2 SAS

SAS はワンタイムパスワード認証方式の 1 種であり，中間者攻撃によるなりすましやリプレイアタックに対して耐性を持っている方式である．SAS は，一方向性関数と排他的論理和によって構成されている．SAS には，SAS-2[7]，SAS-X，SAS-L など複数の種類が存在する．以下に，既存方式およびで用いている SAS-X および提案方式で利用している SAS-L の説明を行う．

2.2.1 SAS-X

SAS-X は，SAS-2 の問題点であったサーバが保存している認証情報が漏洩した場合になりすまされてしまう点を解決した方式である．すなわち，サーバが保存している認証情報が漏洩し盗聴された場合においても，なりすまされることなく認証が可能である方式である．SAS-X はユーザが初回認証情報を生成・送信し，サーバがそれを保存してユーザの登録を行う登録フェーズと，ユーザが算出した認証情報の正当性をサーバが検証し，サーバが算出した認証情報の正当性をユーザが検証することで相互認証を行う認証フェーズの 2 フェーズから構成される．以下に各フェーズの説明を行う．

定義と記法

SAS-X の説明で用いる定義と記法は以下である．

- ユーザは，認証される被認証者を示す．
- サーバは，ユーザを認証する認証者を示す．
- ID は，ユーザの識別子を示す．
- S は，ユーザのパスワードを示す．
- i は，i 回目の認証セッションを示す．
- N_i は，i 回目の認証時に生成される乱数を示す．

- A_i，F_i は，i 回目の認証情報を示す．
- H は，一方向性関数を示す．例として $H(x)$ は x に一方向性関数を適用して得た出力値を示す．
- $\oplus$ は，排他的論理和演算子を示す．

登録フェーズ

図 2.1 に SAS-X の登録フェーズを示す．また，登録の手順を以下に示す．

1. ユーザは自身の識別子 ID とパスワード S を入力する．
2. ユーザは乱数 N_1 を生成し，保存する．
3. ユーザは乱数 N_1 と入力したパスワード S を用いて $A_1 = H(N_1 \oplus S)$ を算出し，この A_1 を用いて初回認証情報 $F_1 = H(A_1)$ を算出する．
4. ユーザは ID と F_1 を安全なルートを用いてサーバへ送信する．
5. サーバは送信されてきた ID と F_1 を保存する．

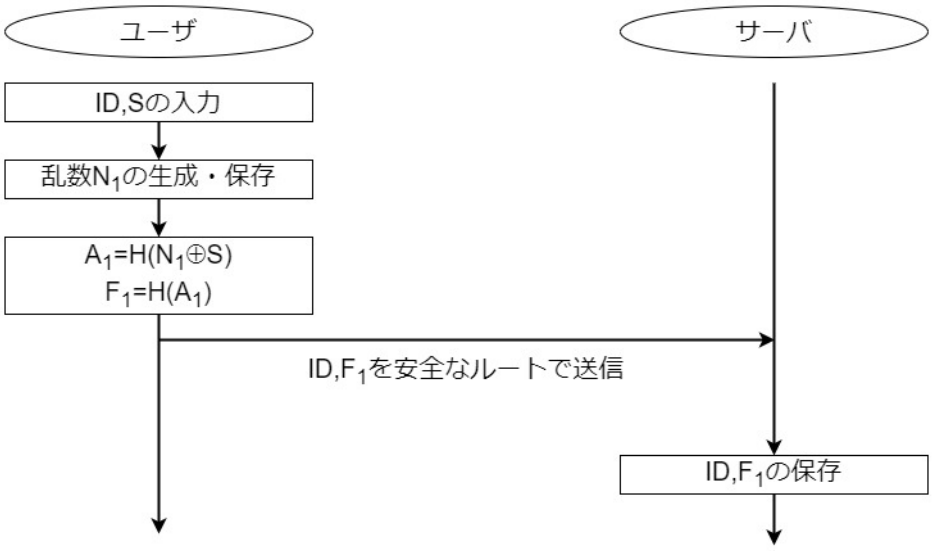


図 2.1 SAS-X: 登録フェーズ

認証フェーズ

あらかじめユーザは乱数 N_i を保持しており，サーバは認証情報 F_i を保持しているものとする．図 2.2 に SAS-X の認証フェーズを示す．また，認証の手順を以下に示す．

1. ユーザは自身の識別子 ID とパスワード S を入力する．
2. ユーザは保持している N_i と入力した S を用いて $A_i = H(N_i \oplus S)$ を算出し，この A_i を用いて認証情報 $F_i = H(A_i)$ を算出する．
3. ユーザは新たな乱数 N_{i+1} を生成し，保存する．
4. ユーザは乱数 N_{i+1} と S を用いて $A_{i+1} = H(N_{i+1} \oplus S)$ を算出し，この A_{i+1} を用いて次回認証情報 $F_{i+1} = H(A_{i+1})$ を算出する．
5. ユーザは F_i と F_{i+1}，A_i を用いて，$\alpha = F_{i+1} \oplus F_i$ と $\beta = F_{i+1} \oplus A_i$ を算出する．
6. ユーザは ID，α，β をサーバへ送信する．この時の経路はインターネット等の安全なルートでなくてもよい．
7. サーバは送信されてきた α に対して保持している F_i を用いて，$F_{i+1} = \alpha \oplus F_i$ を算出し，この F_{i+1} と送信されてきた β から $A_i = \beta \oplus F_{i+1}$ を算出する．
8. サーバは保持している F_i と $H(A_i)$ を比較し，ユーザの認証を行う．値が等しい場合は認証成立とし，等しくない場合は認証不成立とする．
9. 認証成立時サーバは次回認証のために認証情報 F_i を F_{i+1} へ更新する．
10. サーバは $\gamma = H(F_{i+1})$ を算出し，γ をユーザへ送信する．
11. ユーザは送信されてきた γ と $H(F_{i+1})$ を比較し，サーバの認証を行う．値が等しい場合は認証成立とし，等しくない場合は認証不成立とする．

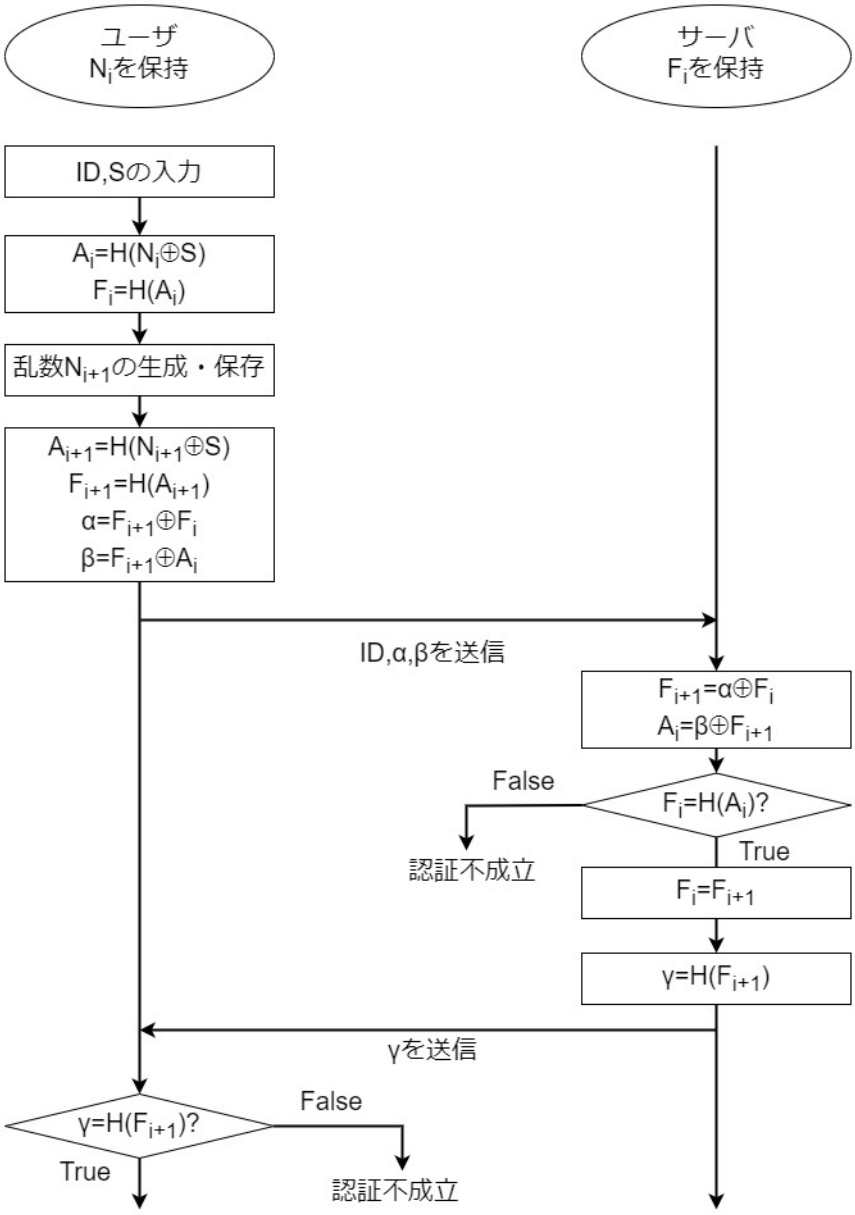


図 2.2 SAS-X: 認証フェーズ

2.2.2 SAS-L

SAS-L は，擬似乱数生成手段 (以下，擬似乱数関数) および一方向性関数が認証側・被認証側の双方で必要である従来方式とは異なり，擬似乱数関数および一方向性関数が認証側・被認証側のいずれかで必要としない方式である [8]．SAS-L には認証側で擬似乱数関数と一方向性関数の適用回数が 0 回である Type1 と，被認証側で擬似乱数関数と一方向性関数の

適用回数が 0 回である Type2 があり，本研究では Type1 を用いている．SAS-L は，ユーザが初回認証情報を生成・送信し，サーバがそれを保存してユーザの登録を行う登録フェーズと，ユーザが算出した認証情報の正当性をサーバが検証しユーザの認証を行う認証フェーズの 2 フェーズから構成される．以下に各フェーズの説明を行う．

定義と記法

SAS-L の説明で用いる定義と記法は以下である．

- ユーザは，認証される被認証者を示す．
- サーバは，ユーザを認証する認証者を示す．
- ID は，ユーザの識別子を示す．
- S は，ユーザのパスワードを示す．
- i は，i 回目の認証セッションを示す．
- N_i は，i 回目の認証時に生成される乱数を示す．
- A_i は，i 回目の認証情報を示す．
- H は，一方向性関数を示す．例として $H(x)$ は x に一方向性関数を適用して得た出力値を示す．
- $\oplus$ は，排他的論理和演算子を示す．
- $+$は，加算演算子を示す．

登録フェーズ

図 2.3 に SAS-L の登録フェーズを示す．また，登録の手順を以下に示す．

1. ユーザは自身の識別子である ID とパスワード S を入力する．
2. ユーザは乱数 N_1 を生成し，保存する．
3. ユーザは乱数 N_1 と入力したパスワード S を用いて初回認証情報 $A_1 = H(N_1 \oplus S)$ を

算出する．

4. ユーザは ID と A_1 を安全なルートを用いてサーバへ送信する．
5. サーバは送信されてきた ID と A_1 を保存する．

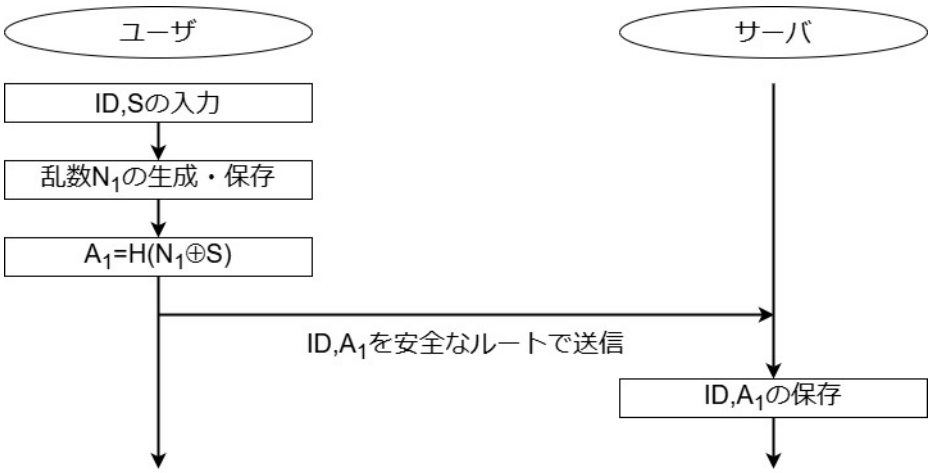


図 2.3 SAS-L: 登録フェーズ

認証フェーズ

あらかじめユーザは乱数 N_i を保持しており，サーバは認証情報 A_i を保持しているものとする．図 2.4 に SAS-L の認証フェーズを示す．また，認証の手順を以下に示す．

1. ユーザは自身の識別子 ID とパスワード S を入力する．
2. ユーザは保持している N_i と入力した S を用いて $A_i = H(N_i \oplus S)$ を算出する．
3. ユーザは新たな乱数 N_{i+1} を生成し，保存する．
4. ユーザは乱数 N_{i+1} と S を用いて次回認証情報 $A_{i+1} = H(N_{i+1} \oplus S)$ を算出する．
5. ユーザは A_i と A_{i+1} を用いて，$\alpha = A_{i+1} \oplus A_i$ と $\beta = A_{i+1} + A_i$ を算出する．
6. ユーザは ID，α，β をサーバへ送信する．この時の経路はインターネット等の安全なルートでなくてもよい．
7. サーバは送信されてきた α に対して保持している A_i を用いて $A_{i+1} = \alpha \oplus A_i$ を算出し，算出した A_{i+1} と保持している A_i から $B = A_{i+1} + A_i$ を算出する．

8. サーバは算出した B と送信されてきた β を比較し，ユーザの認証を行う．値が等しい場合は認証成立とし，等しくない場合は認証不成立とする．
9. 認証成立時，サーバは次回認証のために認証情報 A_i を A_{i+1} へ更新する．
10. ユーザは次回認証のために乱数 N_i を N_{i+1} へ更新する．

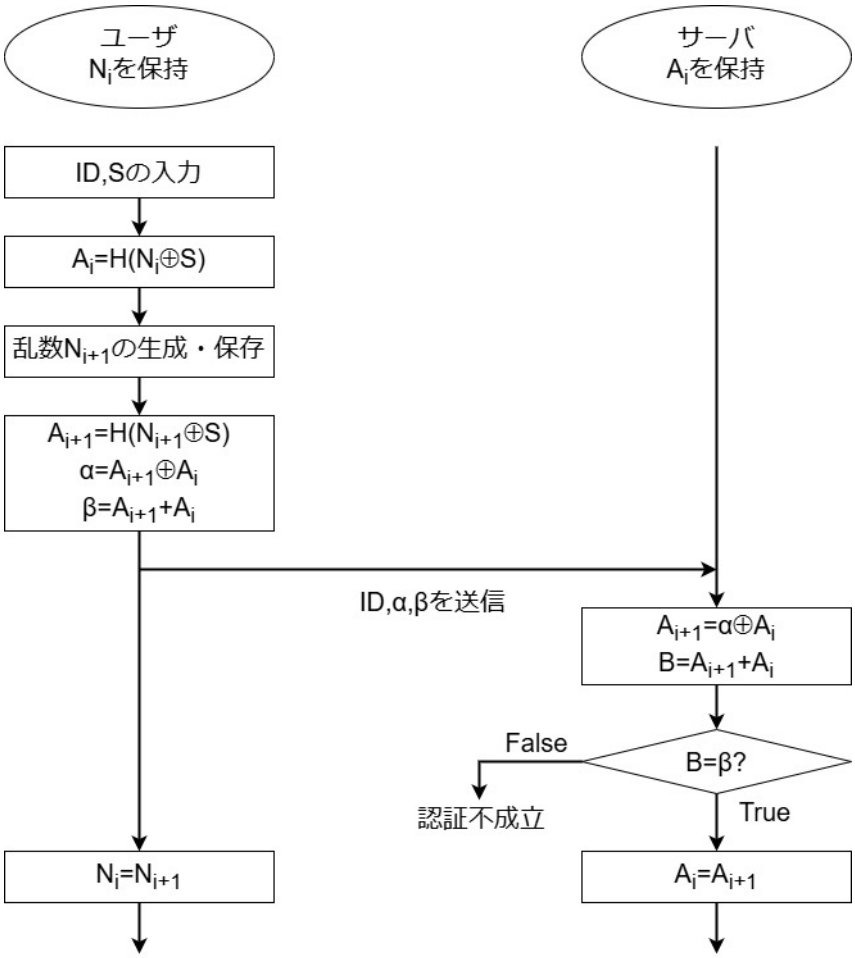


図 2.4　SAS-L:i 回目認証フェーズ

第 3 章

既存方式

複数端末認証が可能なパスワード管理システムとして渡邊の方式 [5]，高橋の方式 [6] がある．以下に，渡邊の方式，高橋の方式の概要と各フェーズの手順，およびこれらの方式の問題点について述べる．

3.1 渡邊の方式

渡邊の方式は，SAS-X を用いることで SSL を用いることなくサーバとの通信を安全に行い，かつ，複数端末の認証を行うことが可能な方式である．この方式は，ユーザが事前にサーバへ認証情報を登録する初期登録フェーズ，ユーザとサーバがともに正規の相手であるかどうか認証を行う初回認証フェーズおよび i 回目認証フェーズ，複数端末から利用できるように新規端末を登録する端末登録フェーズから構成される．以下に，各フェーズの説明を行う．

3.1.1 定義と記法

- ユーザは，認証される被認証者を示す．
- サーバは，ユーザを認証する認証者を示し，ユーザの登録する情報を管理する．
- 端末 A，端末 B は，別のスマートデバイスを指す．
- ID は，ユーザの識別子を示す．
- PW は，ユーザのパスワードを示す．
- ID_G は，サーバに保存しているグループ ID を示す．

- i は，i 回目の認証セッションを示す．
- N_i は，i 回目の認証時に生成される乱数を示す．
- A_i，F_i，G，GSA は，認証情報を示す．なお，添え字に i がつくものは i 回目の認証情報であることを示す．
- SI は，共有情報を示す．
- H は，一方向性関数を示す．例として $H(x)$ は x に一方向性関数を適用して得た出力値を示す．
- $\oplus$ は，排他的論理和演算子を示す．

3.1.2　初期登録フェーズ

図 3.1 に渡邊の方式の初期登録フェーズを示す．また，登録の手順を以下に示す．

1. ユーザは自身の識別子である ID とパスワード PW を入力する．
2. ユーザは乱数 N_1 を生成し，保存する．
3. ユーザは乱数 N_1 と入力したパスワード PW を用いて $A_1 = H(N_1 \oplus PW)$ を算出し，この A_1 を用いて初回認証情報 $F_1 = H(A_1)$ を算出する．
4. ユーザは ID と F_1 をサーバへ送信する．
5. サーバは送信されてきた ID と F_1 を保存する．

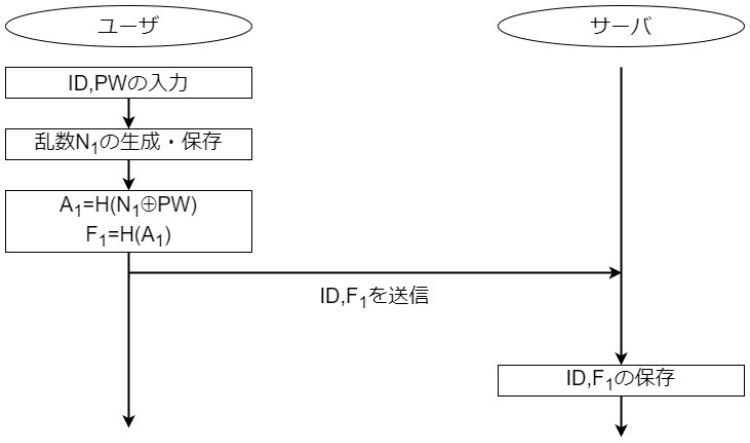


図 3.1　渡邊の方式: 初期登録フェーズ

3.1.3 初回認証フェーズ

あらかじめユーザは乱数 N_1 を保持しており，サーバはユーザの識別子 ID，グループ ID である ID_G，認証情報 F_1 を保持しているものとする．図 3.2 に渡邊の方式の初回認証フェーズを示す．また，認証の手順を以下に示す．

1. ユーザは自身の識別子 ID とパスワード PW を入力する．
2. ユーザは保持している N_i と入力した PW を用いて $A_i = H(N_i \oplus PW)$ を算出し，この A_i を用いて認証情報 $F_i = H(A_i)$ を算出する．
3. ユーザは新たな乱数 N_2 を生成し，保存する．
4. ユーザは乱数 N_2 と PW を用いて $A_2 = H(N_2 \oplus PW)$ を算出し，この A_2 を用いて次回認証情報 $F_2 = H(A_2)$ を算出する．
5. ユーザは F_1 と F_2，A_1 を用いて，$\alpha = F_2 \oplus F_1$ と $\beta = F_2 \oplus A_1$ を算出する．
6. ユーザは ID，α，β をサーバへ送信する．この時の経路はインターネット等の安全なルートでなくてもよい．
7. サーバは送信されてきた α に対して保持している F_1 を用いて，$F_2 = \alpha \oplus F_1$ を算出し，この F_2 と送信されてきた β から $A_1 = \beta \oplus F_2$ を算出する．
8. サーバは保持している F_1 と $H(A_1)$ を比較し，ユーザの認証を行う．値が等しい場合は認証成立とし，等しくない場合は認証不成立とする．
9. 認証成立時，サーバは次回認証のために認証情報 F_1 を F_2 へ更新する．
10. サーバは共有情報 SI と $\gamma = H(F_2)$ を算出する．
11. サーバは SI と γ をユーザへ安全な通信路で送信する．
12. サーバは算出した F_2 と保持している ID_G を用いて，認証情報 $G = F_2 \oplus ID_G$ を算出し，この G を保存する．
13. ユーザは送信されてきた γ と $H(F_2)$ を比較し，サーバの認証を行う．値が等しい場合は認証成立とし，等しくない場合は認証不成立とする．
14. 認証成立時，ユーザは送信されてきた SI を保存する．

3.1　渡邊の方式

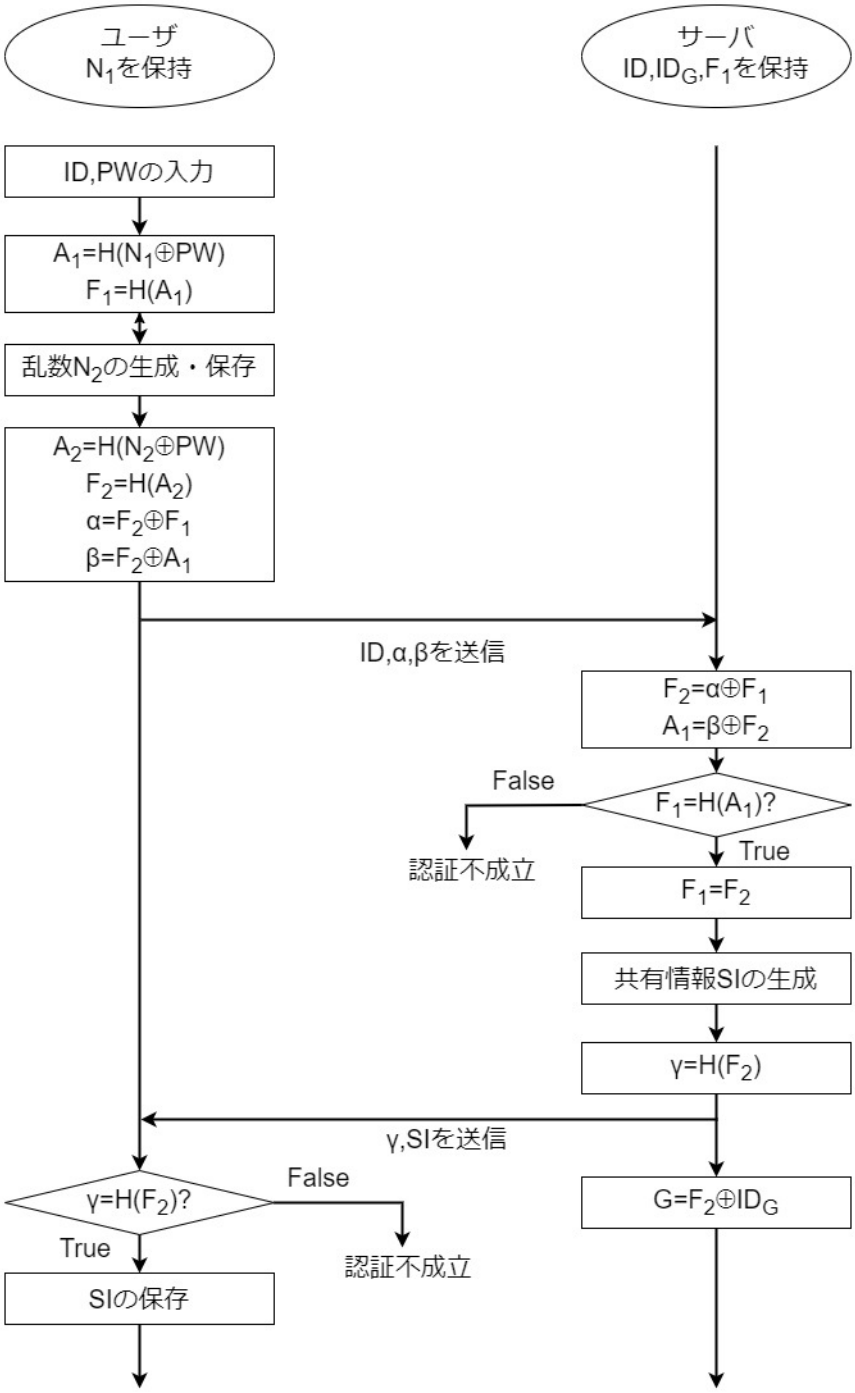


図 3.2　渡邊の方式: 初回認証フェーズ

3.1.4 i 回目認証フェーズ

あらかじめユーザは乱数 N_i，共有情報 SI を保持しており，サーバはユーザの識別子 ID，グループ ID である ID_G，共有情報 SI，認証情報 G を保持しているものとする．図 3.3 に渡邊の方式の i 回目の認証フェーズを示す．また，認証の手順を以下に示す．

1. ユーザはサーバへ認証のリクエストを送信する．
2. サーバはレスポンスとして保持している G をユーザへ送信する．
3. ユーザは自身の識別子 ID とパスワード PW を入力する．
4. ユーザは保持している N_i と入力した PW を用いて $A_i = H(N_i \oplus PW)$ を算出し，この A_i を用いて認証情報 $F_i = H(A_i)$ を算出する．
5. ユーザは新たな乱数 N_{i+1} を生成し，保存する．
6. ユーザは乱数 N_{i+1} と PW を用いて $A_{i+1} = H(N_{i+1} \oplus PW)$ を算出し，この A_{i+1} を用いて次回認証情報 $F_{i+1} = H(A_{i+1})$ を算出する．
7. ユーザは F_i と F_{i+1}，A_i を用いて，$\alpha = F_{i+1} \oplus F_i$ と $\beta = F_{i+1} \oplus A_i$ を算出する．また，送信されてきた G と保持している SI，算出した F_i を用いて，$GSA = G \oplus SI \oplus F_i$ を算出する．
8. ユーザは GSA，ID，α，β をサーバへ送信する．この時の経路はインターネット等の安全なルートでなくてもよい．
9. サーバは送信されてきた GSA に対して保持している G，SI を用いて $F_i = GSA \oplus G \oplus SI$ を算出する．算出した F_i と送信されてきた α から，$F_{i+1} = \alpha \oplus F_i$ を算出し，この F_{i+1} と送信されてきた β から $A_i = \beta \oplus F_{i+1}$ を算出する．
10. サーバは算出した F_i と $H(A_i)$ を比較し，ユーザの認証を行う．値が等しい場合は認証成立とし，等しくない場合は認証不成立とする．
11. 認証成立時，サーバは次回認証のためにユーザの認証情報 F_i を F_{i+1} へ更新する．
12. サーバは新たな共有情報 SI と $\gamma = H(F_{i+1})$ を算出する．
13. サーバは SI と γ をユーザへ安全な通信路で送信する．

14. サーバは算出した F_i，F_{i+1} と保持している G を用いて，認証情報 $G = G \oplus F_i \oplus F_{i+1}$ を算出し，この G を次回認証用に保存する．
15. ユーザは送信されてきた γ と $H(F_{i+1})$ を比較し，サーバの認証を行う．値が等しい場合は認証成立とし，等しくない場合は認証不成立とする．
16. 認証成立時，ユーザは保持している SI を送信されてきた SI に更新する．

3.1　渡邊の方式

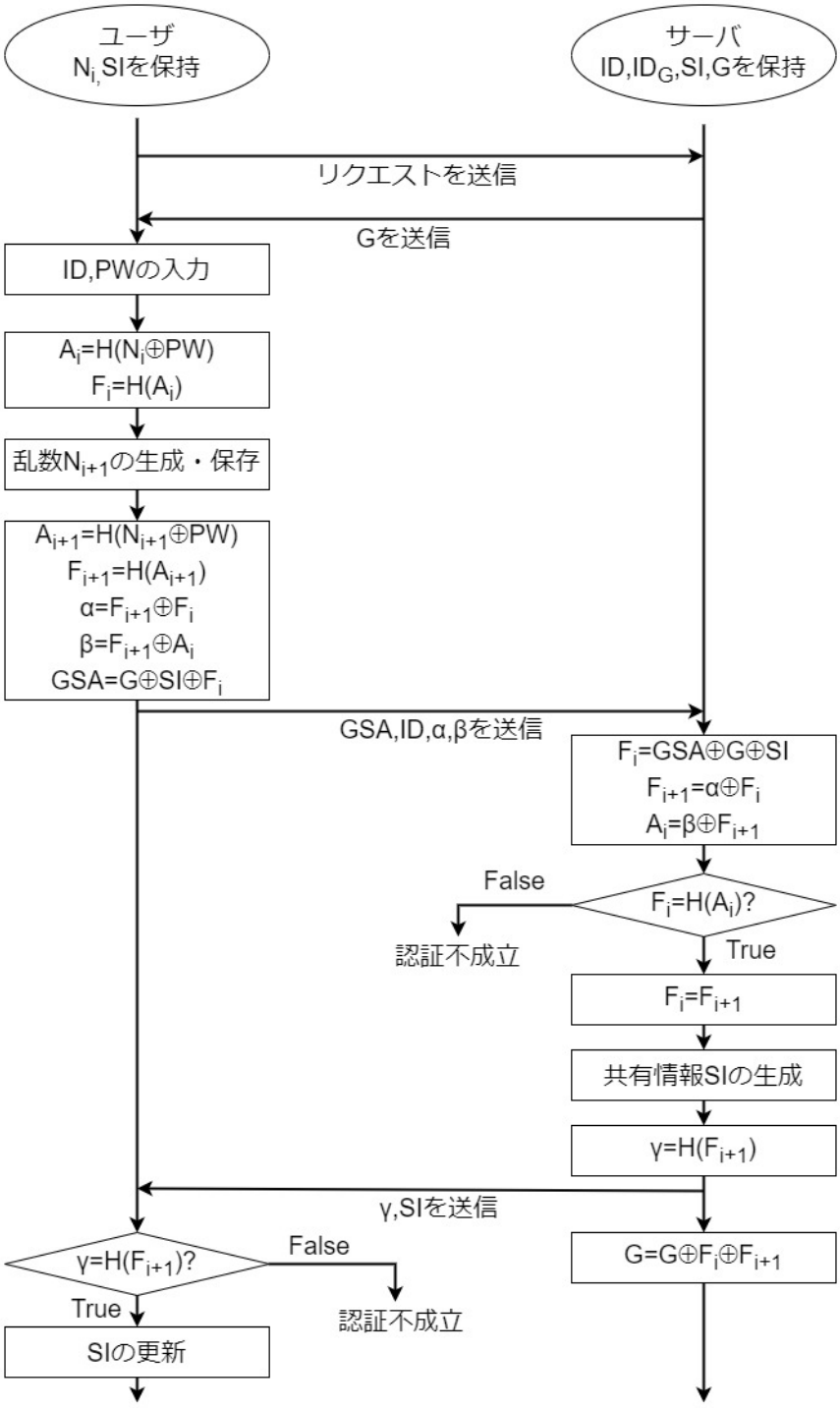


図 3.3　渡邊の方式:i 回目認証フェーズ

3.1.5　端末登録フェーズ

あらかじめサーバは共有情報 SI，認証情報 G を保持しているものとする．図 3.4 渡邊の方式の端末登録フェーズを示す．また，登録の手順を以下に示す．

1. 既に認証済みの端末 A からサーバへ新規端末の登録要求を行う．
2. サーバは登録要求を受け取ると端末登録情報を生成し，これを安全な通信路を用いて端末 A へ送信する．
3. 端末 A は QR コードや $Bluetooth5.3$ 等を用いて端末 B へ端末登録情報を送信する．
4. 端末 B は ID，PW，端末登録情報を入力，初回認証情報 A=を生成し，これらをサーバへ送信する．
5. サーバは送信されてきた端末登録情報を生成したものと比較して認証を行い，認証成功時に端末 B の登録を行う．
6. サーバは $G = G \oplus A$ を算出し，G と SI を更新してユーザに更新後の SI を送信する．
7. ユーザは送信されてきた SI を保存する．

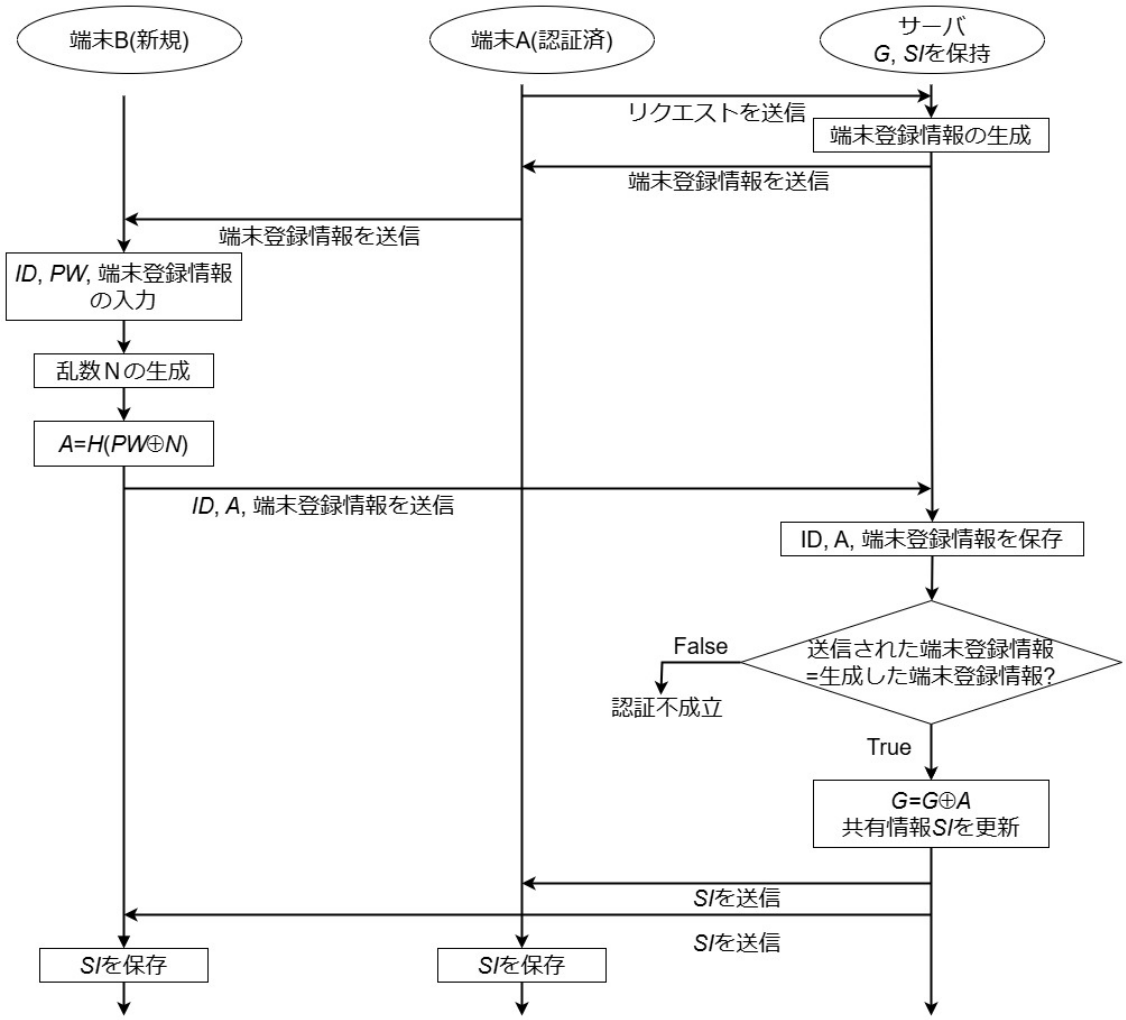


図 3.4　渡邊の方式: 端末登録フェーズ

3.2　高橋の方式

渡邊の方式は，初回認証情報の配送に安全な通信路を必要としない複数端末認証が可能な方式であるが，多端末化する際の新規端末の登録情報の配送には安全な通信路が必要になるという問題点がある．この問題を解決した方式が高橋の方式である．この方式は，システムを利用するユーザの情報をサーバへ登録するユーザ登録フェーズ，ユーザとサーバがともに正規の相手であるかどうか認証を行う i 回目認証フェーズ，複数端末から利用できるように新しい端末をサーバへ登録する新規端末登録フェーズから構成される．以下に，各フェーズの説明を行う．

3.2.1　定義と記法

- ユーザは，システムを利用する利用者を示す.
- サーバは，システムの管理者を示す.
- ID は，ユーザの識別子を示す.
- PW は，ユーザのパスワードを示す.
- i は，i 回目の認証セッションを示す.
- N_i，V_i，K，L は，認証時に生成される乱数を示す．なお，添え字に i がつくものは i 回目の認証で生成される乱数であることを示す.
- A_i，F_i，B_i，D_i は，i 回目の認証情報を示す.
- H は，一方向性関数を示す．例として $H(x)$ は x に一方向性関数を適用して得た出力値を示す.
- $\oplus$ は，排他的論理和演算子を示す.

3.2.2　ユーザ登録フェーズ

図 3.5 に高橋の方式のユーザ登録フェーズを示す．また，登録の手順を以下に示す.

1. ユーザは自身の識別子である ID とパスワード PW を入力する.
2. ユーザは乱数 N_1 を生成し，保存する.
3. ユーザは乱数 N_1 と入力したパスワード PW を用いて $A_1 = H(N_1 \oplus PW)$ を算出し，この A_1 を用いて初回認証情報 $F_1 = H(A_1)$ を算出する.
4. ユーザは ID と F_1 をサーバへ送信する.
5. サーバは送信されてきた ID と F_1 を保存する.

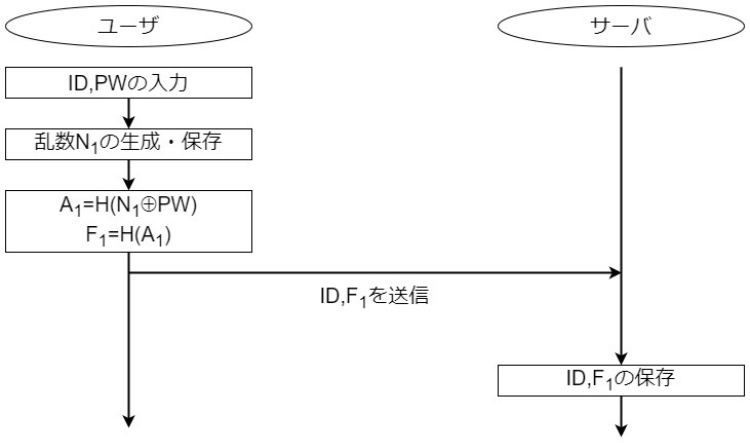


図 3.5　高橋の方式: ユーザ登録フェーズ

3.2.3　i 回目認証フェーズ

あらかじめユーザは乱数 N_i を保持しており，サーバはユーザの識別子 ID，認証情報 F_i を保持しているものとする．図 3.6 に高橋の方式の i 回目の認証フェーズを示す．また，認証の手順を以下に示す.

1. ユーザは自身の識別子 ID とパスワード PW を入力する.
2. ユーザは保持している N_i と入力した PW を用いて $A_i = H(N_i \oplus PW)$ を算出し，この A_i を用いて認証情報 $F_i = H(A_i)$ を算出する.
3. ユーザは新たな乱数 N_{i+1} を生成し，保存する.
4. ユーザは乱数 N_{i+1} と PW を用いて $A_{i+1} = H(N_{i+1} \oplus PW)$ を算出し，この A_{i+1} を用いて次回認証情報 $F_{i+1} = H(A_{i+1})$ を算出する.
5. ユーザは F_i と F_{i+1}，A_i を用いて，$\alpha = F_{i+1} \oplus F_i$ と $\beta = F_{i+1} \oplus A_i$ を算出する.
6. ユーザは ID，α，β をサーバへ送信する．この時の経路はインターネット等の安全なルートでなくてもよい.
7. サーバは送信されてきた α に対して保持している F_i を用いて，$F_{i+1} = \alpha \oplus F_i$ を算出し，この F_{i+1} と送信されてきた β から $A_i = \beta \oplus F_{i+1}$ を算出する.
8. サーバは保持している F_i と $H(A_i)$ を比較し，ユーザの認証を行う．値が等しい場合は

認証成立とし，等しくない場合は認証不成立とする．

9. 認証成立時，サーバは次回認証のために認証情報 F_i を F_{i+1} へ更新する．
10. サーバは $\gamma = H(F_{i+1})$ を算出し，γ をユーザへ送信する．
11. ユーザは送信されてきた γ と $H(F_{i+1})$ を比較し，サーバの認証を行う．値が等しい場合は認証成立とし，等しくない場合は認証不成立とする．

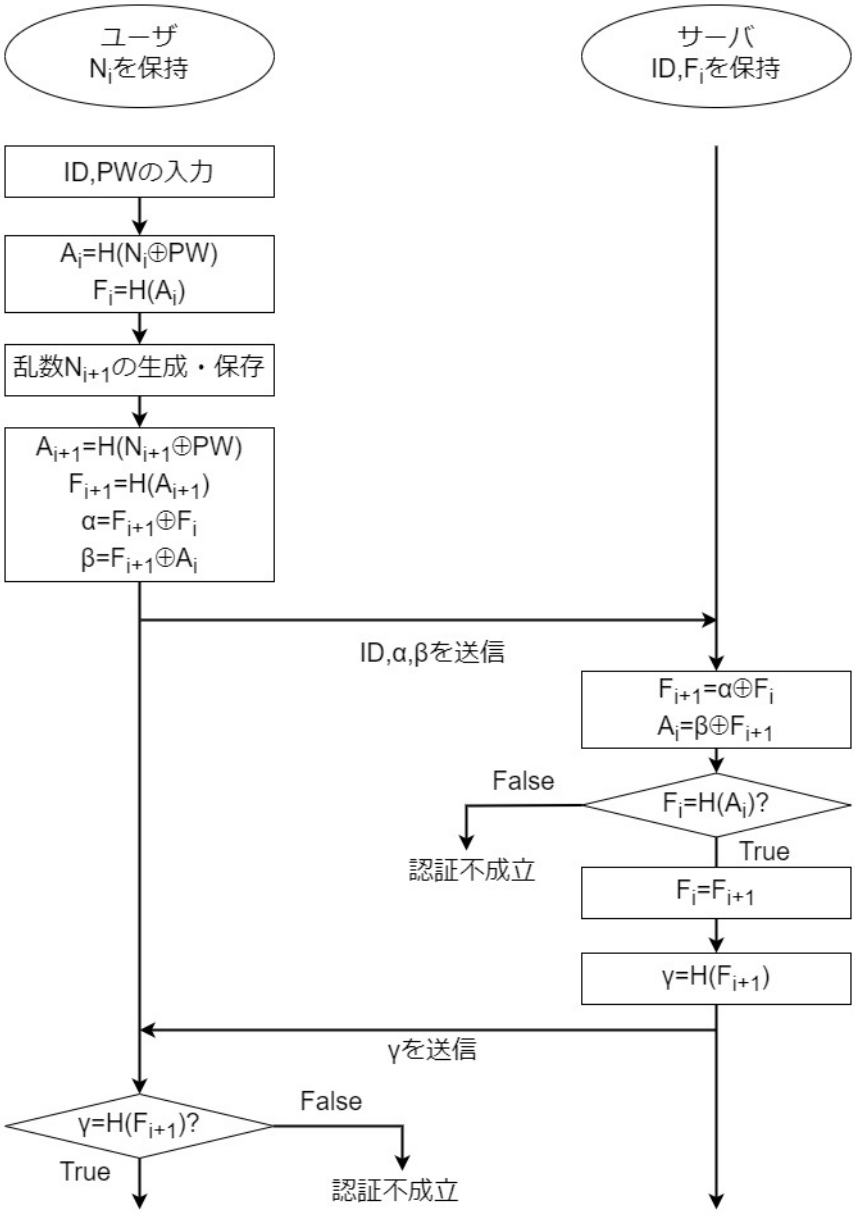


図 3.6　高橋の方式:i 回目認証フェーズ

3.2.4　新規端末登録フェーズ

あらかじめ認証済み端末 A は乱数 N_i を保持しており，サーバはユーザの識別子 ID，認証情報 F_i を保持しているものとする．図 3.7 に高橋の方式の新規端末登録フェーズを以下に示す．また，登録の手順を以下に示す．

1. 認証済み端末 A(以下「端末 A」という) はサーバへ認証のリクエストを送信する．
2. サーバは新たな乱数 K を生成し，保存する．そして，この K を端末 A へ送信する．
3. 端末 A は自身の識別子 ID とパスワード PW を入力する．
4. 端末 A は保持している乱数 N_i と入力した PW を用いて $A_i = H(N_i \oplus PW)$ を算出し，この A_i を用いて認証情報 $F_i = H(A_i)$ を算出する．
5. 端末 A は新たな乱数 N_{i+1} を生成し，保存する．
6. 端末 A は乱数 N_{i+1} と PW を用いて $A_{i+1} = H(N_{i+1} \oplus PW)$ を算出し，この A_{i+1} を用いて次回認証情報 $F_{i+1} = H(A_{i+1})$ を算出する．
7. 端末 A は算出した F_{i+1} と送信されてきた K を用いて，新規登録端末 B(以下「端末 B」という) の認証情報 $B_i = H(F_{i+1} \oplus K)$ を算出し，この B_i を端末 B へ送信する．
8. 端末 B は自身の識別子 ID とパスワード PW を入力する．
9. 端末 B は新たな乱数 L，V_i を生成・保存する．
10. 端末 B は入力した ID，PW，生成した K，V_i，送信されてきた B_i を用いて $D_i = H(B_i)$，$B_{i+1} = H(PW \oplus V_i)$，$ID_B = H(ID \oplus L)$ を算出し，この B_{i+1} を用いて次回認証情報 $D_{i+1} = H(B_{i+1})$ を算出する．
11. 端末 B は送信されてきた B_i と算出した D_i，D_{i+1} から $\alpha_B = D_{i+1} \oplus D_i$ と $\beta_B = D_{i+1} \oplus B_i$ を算出する．
12. 端末 B は ID，ID_B，α_B，β_B をサーバへ送信する．
13. サーバは端末 B から送信されてきた情報を一時的に保存する．
14. 端末 B はサーバへ情報を送信した旨を端末 A へ通知する．
15. 端末 A は端末 B からの通知を受け取ると，算出した A_i，F_i，F_{i+1} を用いて $\alpha_A =$

$F_{i+1} \oplus F_i$ と $\beta_A = F_{i+1} \oplus A_i$ を算出する．

16. 端末 A は ID，α_A，β_A をサーバへ送信する．
17. サーバは送信されてきた α_A に対して保持している F_i を用いて，$F_{i+1} = \alpha_A \oplus F_i$ を算出し，この F_{i+1} と送信されてきた β_A から $A_i = \beta_A \oplus F_{i+1}$ を算出する．
18. サーバは保持している F_i と $H(A_i)$ を比較し，端末 A の認証を行う．値が等しい場合は認証成立とし，等しくない場合は認証不成立とする．
19. 認証成立時，サーバは端末 A の認証情報 F_i を F_{i+1} へ更新する．
20. サーバは $\gamma_A = H(F_{i+1})$ を算出し，γ_A を端末 A へ送信する．
21. 端末 A は送信されてきた γ_A と $H(F_{i+1})$ を比較し，サーバの認証を行う．値が等しい場合は認証成立とし，等しくない場合は認証不成立とする．
22. サーバは更新した端末 A の認証情報 F_{i+1} と保持している乱数 K で端末 B の認証情報 $B_i = H(F_{i+1} \oplus K)$ を算出する．
23. サーバは送信されてきた β_B に対して算出した B_i を用いて，$D_{i+1} = \beta_B \oplus B_i$ を算出し，この D_{i+1} と送信されてきた α_B から $D_i = \alpha_B \oplus D_{i+1}$ を算出する．
24. サーバは算出した D_i と $H(B_i)$ を比較し，端末 B の認証を行う．値が等しい場合は認証成立とし，等しくない場合は認証不成立とする．
25. 認証成立時，サーバは端末 B の認証情報 D_i を D_{i+1} へ更新して端末 B を登録する．なお，登録の際には送信されてきた ID と同じユーザが端末を利用しているものとして，ID と ID_B をグループ化して登録しておく．
26. サーバは $\gamma_B = H(D_{i+1})$ を算出し，γ_B を端末 B へ送信する．
27. 端末 B は送信されてきた γ_B と $H(D_{i+1})$ を比較し，サーバの認証を行う．値が等しい場合は認証成立とし，等しくない場合は認証不成立とする．

3.2　高橋の方式

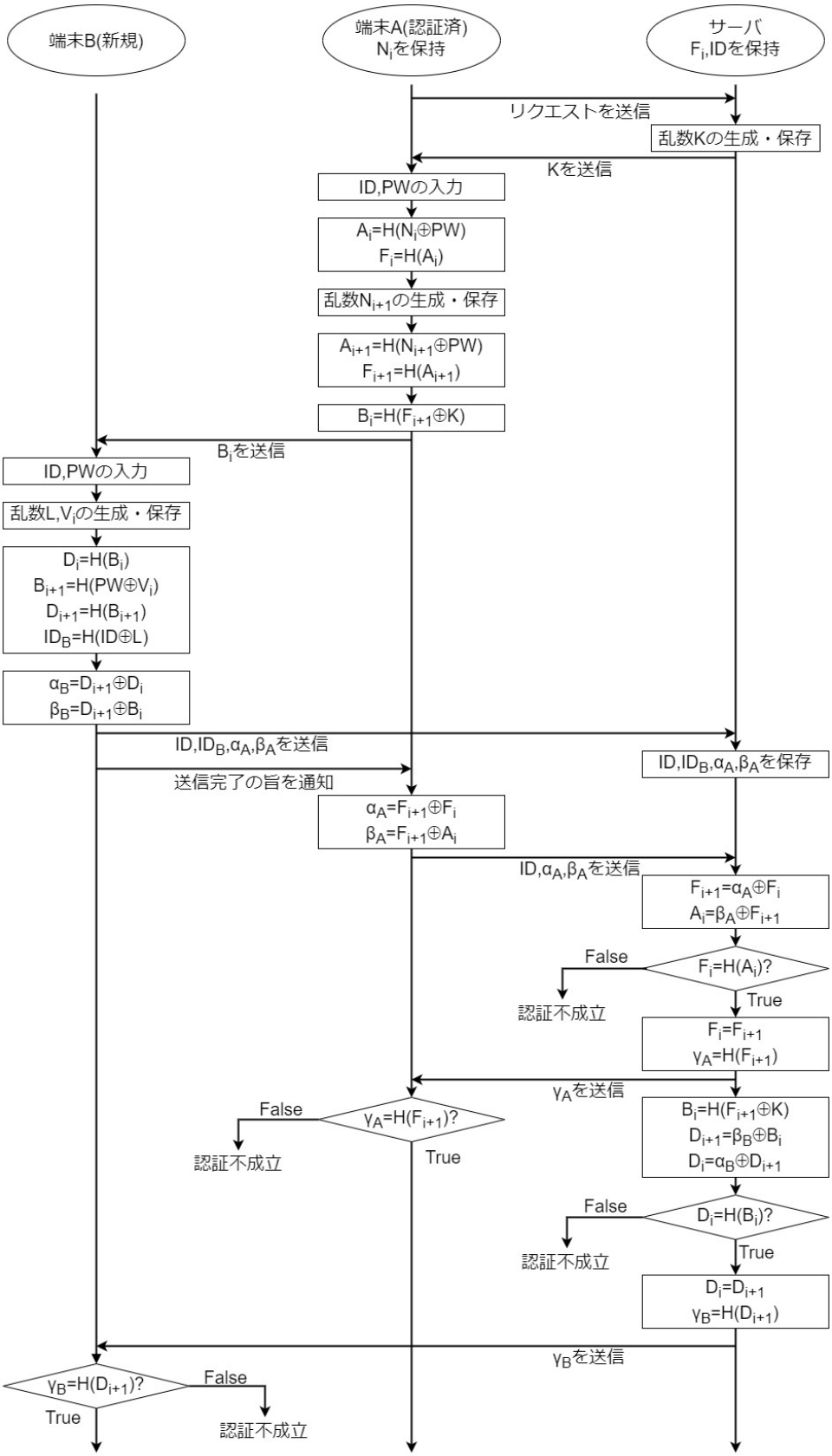


図 3.7　高橋の方式: 新規端末登録フェーズ

3.3　既存方式の問題点

渡邊の方式，高橋の方式はどちらも複数端末認証が可能なパスワード管理システムの方式ではあるが，新規端末の登録が成功した際にユーザの端末ごとに認証情報を管理する必要がある．また，新規端末の登録の際には新規端末の識別子 ID を既に登録済みの端末と同じユーザが利用しているものとして管理するために，渡邊の方式では共有情報 SI を端末ごとに保持する必要がある．高橋の方式ではその ID をグループ化して登録しておく必要がある．したがって，既存方式では利用するユーザの人数に対して管理する必要のある認証情報の数が圧倒的に多く，ワンタイムパスワードを効率よく利用できていない問題点が挙げられる．

第 4 章

提案方式

提案方式では複数端末認証が可能であり，既存方式で示した課題であるワンタイムパスワードを効率的に利用できていない点を解決している．提案方式は既存方式とは異なり認証には SAS-L を用いる．また，認証に用いる乱数とインクリメント回数をサーバ側で生成・保持し，ユーザは認証ごとにサーバから送信される乱数とインクリメント回数を用いる形で認証が進む．このとき，乱数とインクリメント回数はサーバのみが保持していることでユーザの端末による差異がなくなるため，認証情報はユーザ単位で管理するだけでよくなる．

提案方式はユーザの認証情報および利用しているサービス名，ID とパスワードの組を事前に共有・登録する登録フェーズとサーバがユーザの認証を行い，認証成功時にパスワードを取得することができる利用フェーズの 2 フェーズから構成される．以下に，各フェーズの説明を行う．

4.1 定義と記法

- ユーザは，システムを利用する利用者を示す.
- サーバは，システムの管理者を示す.
- S は，ユーザの秘密のパスワードを示す.
- N は，ユーザが利用しているサービス名を示す.
- ID は，ユーザが利用しているサービスの識別子を示す.
- PW は，ユーザが利用しているサービスのパスワードを示す.
- i は，i 回目の認証セッションを示す.

- R は，サーバが生成する乱数を示す．
- n は，インクリメント回数を示す．
- A_i は，i 回目の認証情報を示す．
- H は，一方向性関数を示す．例として $H(x)$ は x に一方向性関数を適用して得た出力値を示す．
- E は，暗号化を示す．例として $E(x)$ は x を暗号化したものを示す．
- D は，復号を示す．例として $D(x)$ は x を復号したものを示す．
- $\oplus$ は，排他的論理和演算子を示す．
- $+$は，加算演算子を示す．

4.2　登録フェーズ

図 4.1 に提案方式の登録フェーズを示す．また，その手順を以下に示す．

1. ユーザは秘密のパスワード S を入力する．
2. ユーザはサーバへ登録要求を送信する．
3. サーバは乱数 R とインクリメント回数 n を生成・保存し，ユーザへこれらを送信する．
4. ユーザは入力した S と送信されてきた R と n を用いて初回認証情報 $A_1 = H(S \oplus (R + n))$ を算出する．
5. ユーザは算出した A_1 を安全なルートを用いて送信する．
6. サーバは送信されてきた A_1 を保存する．
7. ユーザは自身が利用しているサービス名 N，サービスの識別子 ID，サービスのパスワード P の組を入力する．その後，S を暗号鍵として，入力した ID と P の組を暗号化した $Data = E((ID, P), S)$ を生成する．
8. ユーザは生成した $Data$ と N を対応付けてサーバへ送信する．なお，ユーザは登録したいサービスの数だけ 7. と 8. の操作を繰り返す．
9. サーバはユーザから送信されてきた $Data$ と N を対応付けて保存する．

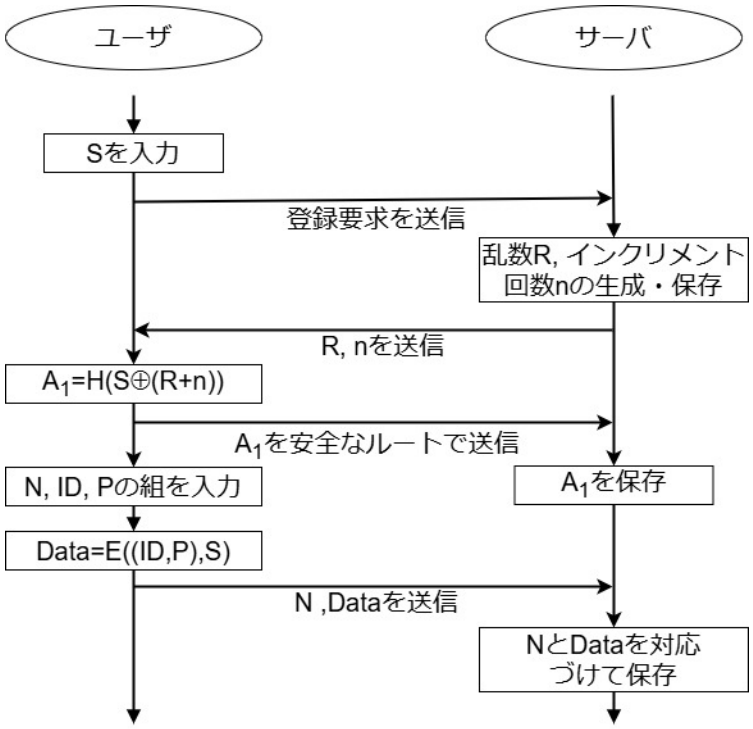


図 4.1　提案方式: 登録フェーズ

4.3　利用フェーズ

あらかじめユーザは秘密のパスワード S を保持しており，サーバは乱数 R，インクリメント回数 n，認証情報 A_i，ユーザが登録した N と暗号化された $Data$ の組を複数保持しているものとする．図 4.2 に提案方式の利用フェーズを示す．また，その手順を以下に示す．

1. ユーザはサーバへ認証要求を送信する．
2. サーバはユーザへ乱数 R とインクリメント回数 n を送信する．
3. ユーザはパスワードを知りたいサービス名 N を入力する．
4. ユーザは保持している S と送信されてきた R と n を用いて $A_i = H(S \oplus (R+n))$ を算出する．
5. ユーザはインクリメント回数 n に 1 を足した n+1 を用いて $A_{i+1} = H(S \oplus (R+(n+1)))$ を算出する．
6. ユーザは算出した A_{i+1} と A_i を用いて，$\alpha = A_{i+1} \oplus A_i$ と $\beta = A_{i+1} + A_i$ を算出する．

7. ユーザは算出した A_{i+1} を保存する．
8. ユーザは入力した N，算出した α，β をサーバへ送信する．
9. サーバは送信されてきた α に対して保持している A_i を用いて $A_{i+1} = \alpha \oplus A_i$ を算出し，算出した A_{i+1} と保持している A_i から $B = A_{i+1} + A_i$ を算出する..
10. サーバは算出した B と送信されてきた β を比較し，ユーザの認証を行う．値が一致すれば認証成立とし，一致しなければ認証不成立とする．
11. 認証成立時，サーバはインクリメント回数 n を $n + 1$ に，認証情報 A_i を A_{i+1} に更新する．
12. サーバは送信されてきた N に対応する $Data$ を探し，存在すれば $Data$ を取得し，存在しなければエラーメッセージを生成する．
13. サーバはユーザへ取得した $Data$ あるいはエラーメッセージを送信する．
14. ユーザは送信されてきたものがエラーメッセージであればそれを出力する．送信されてきたものが $Data$ であれば S を復号鍵として復号した $(ID, P) = D(Data, S)$ を算出し，ID と P を取得する．

4.3　利用フェーズ

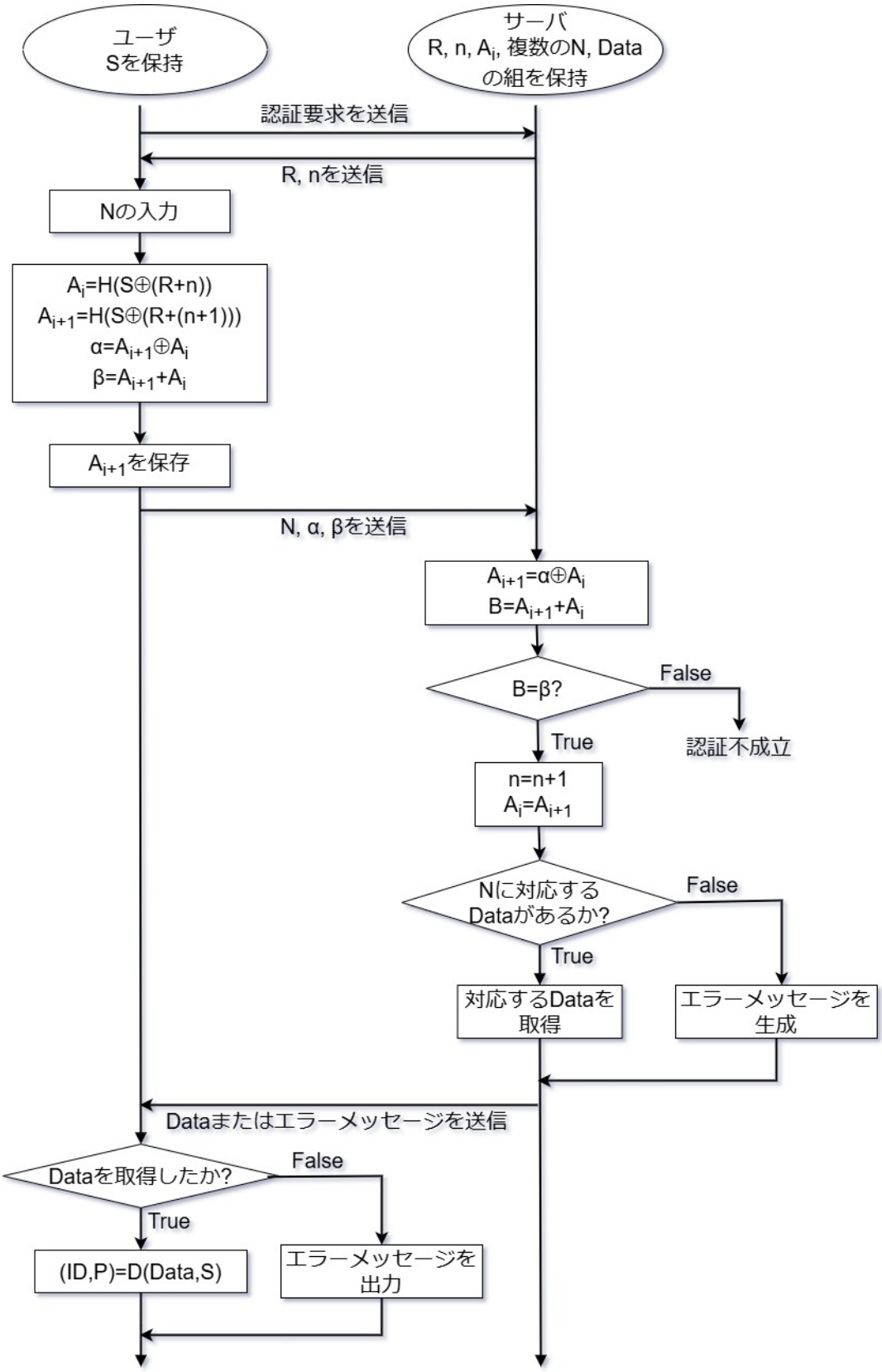


図 4.2　提案方式: 利用フェーズ

第 5 章

評価

既存方式である渡邊の方式，高橋の方式および SAS-L を用いたパスワード管理システムである提案方式を以下の項目で比較する．

1. パスワードの安全性
2. 認証情報管理コスト
3. 認証に必要な通信回数
4. 一方向性関数の適用回数

表 5.1 はパスワードの安全性，認証情報管理コスト，認証に必要な通信回数の 3 項目を比較したものである．また，表 5.2 は一方向性関数の適用回数を初回認証時，同端末で連続して利用した場合の i 回目認証時 (以降，同端末認証時)，ある端末で利用したのちに別の端末で利用した場合の i 回目認証時 (以降，別端末認証時) の 3 通りにおいて比較したものである．

表 5.1　既存方式と提案方式の比較

方式	パスワードの安全性	認証情報管理コスト	認証に必要な通信回数
渡邊の方式	安全	高い	4
高橋の方式	安全	高い	2
提案方式	安全	低い	3

表 5.2　一方向性関数の適用回数の比較

方式	一方向性関数の適用回数					
	初回認証時		i 回目認証時 (同端末)		i 回目認証時 (別端末)	
	ユーザ	サーバ	ユーザ	サーバ	ユーザ	サーバ
渡邊の方式	5	2	5	2	5	2
高橋の方式	5	2	5	2	5	2
提案方式	2	0	1	0	2	0

5.1　パスワードの安全性

渡邊の方式，高橋の方式，提案方式のどの方式においてもワンタイムパスワード認証方式である SAS-X あるいは SAS-L を用いており，通信路上のデータを盗聴された場合であっても正規のユーザになりすますことができないため安全であるといえる．

5.2　認証情報管理コスト

渡邊の方式，高橋の方式ではユーザ側で乱数を生成する都合上，ユーザの端末ごとに異なる乱数が生成される．このとき，乱数を用いて生成する認証情報もユーザの端末ごとに異なりそれらをすべて管理する必要がある．そのため，サーバは認証情報をユーザの端末ごとに管理する必要があり認証情報の管理コストが高くなってしまっている．

提案方式では，認証に必要な R と n はサーバ側が保持しており，S もユーザ単位のものであることからユーザの端末を区別して管理する必要がなくなる．そのため，サーバは認証情報をユーザごとに管理するだけでよくなり認証情報の管理コストが低くなる．

5.3　認証に必要な通信回数

渡邊の方式では，ユーザがサーバにリクエストを送信しレスポンスとして G が返ってくる 2 回と，GSA，ID，α，β の送信と γ の受信の 2 回を合算して計 4 回である．

高橋の方式では，通常の SAS-X と同様の認証フェーズであるため ID，α，β の送信と γ の受信で計 2 回である．

提案方式では，ユーザがサーバに認証要求を送信しレスポンスとして R と n が返ってくる 2 回と，N，α，β を送信する 1 回を合算して計 3 回となっている．高橋の方式と比較して通信回数が増加している要因としては，認証情報の生成に必要な R と n がサーバのみ保持しており，これらの値を取得する通信を追加しなければならないためである．

5.4　一方向性関数の適用回数

渡邊の方式，高橋の方式では一方向性関数をどの認証時においてもユーザ側で 5 回，サーバ側で 2 回の計 7 回適用される．

提案方式では，初回認証時は通常の SAS-L と同じ流れであるため一方向性関数はユーザ側で 2 回，サーバ側で 0 回適用される．同端末認証時は前回認証時に保存した A_{i+1} が今回認証時の A_i として使用することができ，A_i を生成する必要がなくなるため一方向性関数はユーザ側で 1 回，サーバ側で 0 回適用される．別端末認証時では前回認証時に保存した A_{i+1} は今回認証時の A_i として使用することができないため一方向性関数は初回認証時と同じくユーザ側で 2 回，サーバ側で 0 回適用される．

第 6 章

考察

評価より，提案方式は認証情報の管理コストの面および一方向性関数の適用回数の面で既存方式より有用であることを示した．認証情報の管理コストが低いことから提案方式は認証情報の管理が容易となり，結果として認証プロセスの効率化が図られると考えられる．したがって，提案方式は既存方式と比べてワンタイムパスワードを効率的に利用できているといえる．また，一方向性関数の適用回数がサーバ側で 0 回であることから，システムのユーザが増大した場合でもサーバの計算負荷が抑えられると考えられる。

今後の課題として，既存方式との速度評価およびユーザ数の増加に伴うサーバにかかる処理負荷の定量的な比較が挙げられる．また，本研究では SAS-L の通信における同期ずれを考慮していないため，同期ずれが原因となる認証失敗が起こる可能性が考えられる．今後は，実装による定量的評価を行うとともに同期ずれを解決する手法の検討が必要になる．

第 7 章

まとめ

インターネットの普及に伴い Web サービスはその種類も数も増加している．それに比例してユーザ個人が管理する必要のある ID とパスワードの組も増加している．しかし，記憶しておける数には限界がある点や，同じものを使いまわす場合セキュリティが低くなってしまう．これらの対策として，パスワード管理システムがある．しかし，その多くは固定パスワードによる認証が採用され，これは固定パスワードが漏洩すると保存している ID とパスワードの組も漏洩するため，不正利用される危険性が挙げられる．また，パスワード管理システムの従来の方式では公開鍵系の暗号方式を用いた鍵配送の方式があるが，これは鍵交換に時間がかかるという課題がある．これらの課題を解決する方法としてワンタイムパスワード認証方式に基づいた鍵配送の方式が提案されているが，ワンタイムパスワードは通信ごとに認証情報が変化し鍵の更新が困難であるため，複数端末での利用が困難である．この課題の解決には渡邊の方式，高橋の方式などが提案されているが，どちらの方式も認証情報の管理コストが高くなり効率的にワンタイムパスワードを利用できていない．そこで本研究では，既存方式の問題点を解決した複数端末認証が可能なワンタイムパスワードを効率よく利用する方式を提案する．既存方式と比較して，提案方式は認証情報の管理コストおよび一方向性関数の適用回数の面で有用性を示し，ワンタイムパスワードを効率よく利用できていることを示した．

今後の課題として，既存方式との速度評価の比較およびユーザ数の増加に伴うサーバにかかる処理負荷の比較が挙げられる．また，本研究では SAS-L の通信における同期ずれを考慮していないため，同期ずれが原因となる認証失敗が起こる可能性が考えられる．今後は，実装による評価を行うとともに同期ずれを解決する手法の検討が必要になる．

謝辞

本研究と論文作成にあたり，言葉では言い表せないほどの御指導・御助言をいただきました高知工科大学情報学群 清水明宏教授に心より感謝し厚く御礼申し上げます.

また，本研究の副査を担当していただいた高知工科大学情報学群 敷田幹文教授，横山和俊教授に深く感謝申し上げます.

最後に，有益な議論を交わしていただいた高知工科大学 清水研究室の関係者各位に深く感謝いたします.

参考文献

[1] 総務省，”令和 6 年版情報通信白書 -インターネット”，2024，https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/r04/html/nd238120.html

[2] 総務省，”令和 5 年通信利用動向調査の結果”，2024 年 6 月 7 日，https://www.soumu.go.jp/johotsusintokei/statistics/data/240607_1.pdf

[3] トレンドマイクロ株式会社，”パスワードの利用実態調査 2023”，2023 年 8 月 31 日，https://www.trendmicro.com/ja_jp/about/press-release/2023/pr-20230831-01.html

[4] 総務省，”令和 6 年版情報通信白書 -情報通信機器・端末”，2024，https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/r06/html/nd21b110.html

[5] 渡邊 真悟，”SAS-X を用いた複数端末認証可能なパスワード管理システムの提案”，高知工科大学修士学位論文，2017.

[6] 高橋 錬，”サーバと端末間に安全な通信路を必要としない複数端末化可能なパスワード管理システムの提案”，高知工科大学学士学位論文，2018.

[7] T.Tsuji，A.Shimizu，“ A one-time password authentication method for low spec machines and on internet protocols ”，IEICE Trans.Commun.，vol.E87-B，no.6，pp.1594-1600，2004.

[8] K.Mizoguchi，A.Shimizu，”A One-Time Password Authentication Scheme for IoT Environments”，IEICE Technical Report，vol.124，no.257，pp.112-117，2024.