平成 12 年度

学士学位論文

コンテンツ転送サービス方式

Contents Forwarding Services

1010375　　岡崎　友輝年

指導教員　　清水　明宏

2001 年 2 月 5 日

高知工科大学 情報システム工学科

要 旨

コンテンツ転送サービス方式

岡崎　友輝年

出先から，インターネットに接続された中継サーバを経由して，安全に自分宛の電子メールを送受信したり，ファイル転送が可能なサービス方式を提案する．本方式では，ファイル転送に電子メール転送プロトコル SMTP (Simple Mail Transfer Protocol) を用いることによりファイアウォールを回避している．また，安全性を確保するために，SAS-K (SAS - KUT version) 認証方式を用いた 3 者間認証を考案し，ファイル操作に必要な要求情報を暗号化すると共に認証情報を含ませ，第 3 者によるなりすまし行為を不可能とする．

本稿では，内部サーバ，中継サーバ，クライアントの 3 者間で，極めて安全にファイル転送を行なうためのサービス方式を示し，その有効性について述べる．

キーワード　コンテンツ，ファイル，転送，認証，SAS-K，FEAL

Abstract

Contents Forwarding Services

Yukitoshi OKAZAKI

I propose the service that makes possible for you to handle incoming , outgoing mail and transfer files remotely and securely via server which is connected to internet. By this service it is possible to break the firewall for using SMTP when files are transferred. Using SAS-K system between three parties make the file transfer more secured. This system encrypt the necessary information and prevents someone for hacking your data.

In this section, I explain more about this service including its effectiveness and how to transfer files extremely securely between local server, remote server, and client.

key words Contents, File, Forwarding, Authenticstion, SAS-K, FEAL

目次

第 1 章　はじめに　1

第 2 章　従来の 3 者間電子メール転送サービス方式　2

2.1　暗号化機能を有する 3 者間電子メール転送サービス方式 2

2.1.1　高速データ暗号アルゴリズム FEAL 4

2.2　PERM 認証方式 4

2.3　問題点 6

第 3 章　コンテンツ転送サービス方式の提案　7

3.1　3 者間での SAS-K 認証方式の利用 8

3.1.1　記号の定義 8

3.1.2　アルゴリズム 9

3.2　3 者間ファイル転送方式 10

3.2.1　機能設計 10

3.2.2　アルゴリズム 12

第 4 章　考察　17

4.1　3 者間での SAS-K 認証方式の利用 17

4.2　ファイルの漏えい防止 18

4.3　ユーザの利便性 18

第 5 章　まとめ　19

謝辞　20

参考文献　21

目次

付録 A　高速データ暗号アルゴリズム FEAL　22

A.1　記号の定義 22

A.2　FEAL の基本構造 22

付録 B　SAS-K（SAS - KUT version）認証方式　27

B.1　記号の定義 27

B.2　SAS-K プロトコル 28

B.2.1　登録フェーズ $(n = 0)$ 28

B.2.2　登録フェーズ $(n = k)$ 28

B.2.3　同期対策 30

図目次

2.1 暗号化電子メール転送サービス方式の構成例 3
2.2 n 回目の PERM 認証 5

3.1 第 3 者によるなりすまし行為 8
3.2 3 者間による SAS-K 認証方式の流れ 9
3.3 3 者間によるファイル操作の流れ 12
3.4 利用端末のファイル操作のフローチャート 14
3.5 中継サーバのファイル操作のフローチャート 15
3.6 内部サーバのファイル操作のフローチャート 16

A.1 FEAL-N の暗号化の処理の流れ 24
A.2 f 関数 25
A.3 FEAL における鍵生成における基本単位 25
A.4 FEAL の鍵生成用 f_K 関数 26

B.1 SAS-K の登録，及び認証フェーズ 29
B.2 カウンタを利用した同期方法 31

第 1 章

はじめに

安価で使いやすいネットワークとしてインターネットが普及した一方で，パーソナルコンピュータ，モバイル端末は発達し，それに伴い，電子メールが新しいコミュニケーションツールとして，その利用者を増加させてきた．今後はさらに，電子メールに限らず，データファイルの送受信など付加価値の高い電子データを安全に取得可能な方式が必要になることは充分に考えられる．特に，ビジネスユーザにとって，イントラネット内に蓄積された電子データを出先において有効に活用できるかが重要になる．

しかし，出先から電子メールの送受信を行なう場合，ユーザは電子メールの送受信可能な端末を確保しなければならない．また出先において，電子メールの送受信が可能な端末を利用できる場合，その端末にユーザ自身のアカウントを有しているとは考えられない．そのため，他者のアカウントを一時的に借用しなければならない．しかし，例えそのような環境が整っていたとしてもセキュリティ上の問題から，モデム経由でのメールサーバへのアクセス拒否や，ファイアウォールによる内部ネットワークへの不特定な IP アドレスのアクセス拒否などが考えられる．また，FTP[*1] (File Transfer Protocol) によるファイル転送もファイアウォールによって利用できない場合がある．

本論文以下では，第 2 章で従来ある技術について述べ，問題点をあげる．第 3 章では，その問題点を解決する方法として 3 者間ファイル転送方式を提案し，機能とアルゴリズムを示す．第 4 章では，提案した方式について安全性を示し，第 5 章でまとめとする．

[*1] ネットワーク上のクライアントとホストコンピュータとの間で，ファイルの転送を行なうためのプロトコル (またはそれを実装したコマンド)．UNIX では，この FTP プロトコルを実装した ftp コマンドが標準で提供される．

第 2 章

従来の 3 者間電子メール転送サービス方式

本章では，従来からある暗号化機能を有する 3 者間電子メール転送サービス方式のサービス構成の例を示し，認証方式として用いられる PERM (Privacy Enhanced Information Reading and Writing Management Protocol) 認証方式の概要と，サービスを実現する上での問題点を明らかにする．

2.1 暗号化機能を有する 3 者間電子メール転送サービス方式

出先からインターネット経由で安全に電子メールの送受信のサービスを受けたいという要求に対し，3 者間電子メール転送サービス方式 [1] [2] が提案されている．このサービス方式は，電子メールの転送を，ユーザがアカウントを有する内部ネットワーク上のメールサーバ，インターネットに接続された外部ネットワーク上の中継サーバ，ユーザの利用端末の 3 者間で行なうサービス方式である．図 2.1 に構成例を示す．

3 者間の通信プロトコルは，それぞれ，メールサーバ (internal server) - 中継サーバ (transferring server) 間に，電子メール転送プロトコル SMTP[*1] (Simple Mail Transfer Protocol) を用い，中継サーバ - 利用端末 (user) 間には， HTTP[*2] (Hyper Text Transfer

[*1] TCP/IP の上位プロトコルで，電子メール送信システム (MTA) で使われるプロトコル．電子メールソフトがメールサーバにメールを送るときや，メールサーバ間のメールのやり取りに使われる．

[*2] Web ブラウザと Web サーバ間において，HTML ファイルなどの文書を転送するために用いられるアプリケーションレベルのプロトコル．サーバ側のポート番号は 80 番が予約されている．

Protocol) を用いる．

しかし，情報伝達手段として，電子メールを用いてより機密性の高い情報なども取り交わされるようになってきており，転送経路上あるいは中継サーバ内において，盗見，改ざんの脅威にさらされている．

この方式では，認証情報の生成および，電子メールの暗号化，復号を行うために，利用端末上の WWW ブラウザにおいて Java Applet と Java Script を利用する．また，認証情報の作成とメールの暗号化，復号を行うための一方向変換関数として FEAL (Fast Data Encipherment Algorithm) [3] [4] [5] を使用する．

暗号化・復号に関しては，電子メールの本文のみを暗号化し，暗号化方式として，共通鍵暗号方式を利用するものである．利用者は予め，使用する鍵をメールサーバに登録しておき，閲覧時に利用端末で同じ鍵を用いて復号することができる．

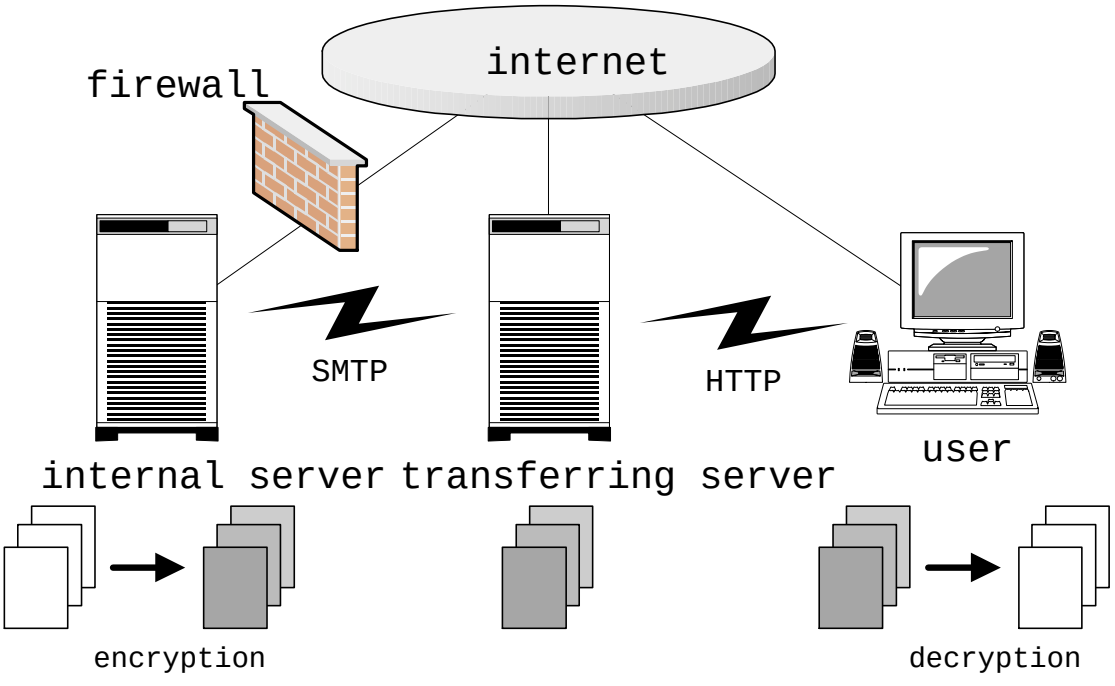

図 2.1　暗号化電子メール転送サービス方式の構成例

2.1.1 高速データ暗号アルゴリズム FEAL

FEAL とはインボリューション構造を用いた共通鍵暗号方式であり，FEAL-NX (128 ビットの鍵を用いる) と FEAL-N (N は 4,8,16,32,…などがあり，暗号化に用いる基本変換単位の段数を表す) の 2 種類がある (付録 A).

主な特徴として，演算型乱数化処理によりプログラムサイズを小さくすることができる，処理単位を 8 bit に統一する事により高速なソフトウェア処理を実現することができる，あみだ状の構造による入力ビット変化伝播効率の向上により高い乱数化効率を得ることができるなどが上げられる.

2.2 PERM 認証方式

暗号化機能を有する 3 者間電子メール転送サービス方式では PERM (Privacy Enhanced Information Reading and Writing Management Protocol) 認証方式 [6] が用いられている．PERM の認証手順においては，認証される側が認証情報を行う認証者に対して，以下の 3 つのデータを送信する.

1. 前回に正当性の確認を終え登録されている一方向変換データの，変換前のデータ，(V)
2. 次々回の認証に用いる一方向変換データ，(W)
3. 次々回の認証に用いる一方向変換データの正当性を次回に確認するための認証子データ，(M)

認証毎に，これらのデータを送信することによって，認証情報を更新しながら，連鎖的に認証を行なうことが可能となる.

内部サーバにおける n 回目の認証処理は，図 2.3 に示すように，以下の手順で行なわれる.

利用者から送信された V_{n-1} と利用者のユーザ ID から算出される一方向変換データと，メールサーバに蓄積され既にその正当性が証明されている W_{n-1} を比較し (1)，V_{n-1} の

正当性を証明する．

次に，利用者から送信され，その正当性が証明された V_{n-1} と，内部サーバに蓄積されているデータ W_n から算出される一方向変換データと，内部サーバに蓄積されている認証子 M_{n-1} とを比較し（2），W_n の正当性を検証する．正当性が証明された W_n は，次回の認証に使用する．認証処理後，その正当性が証明された W_n と今回送付した W_{n+1}，M_n を内部サーバに保存する．

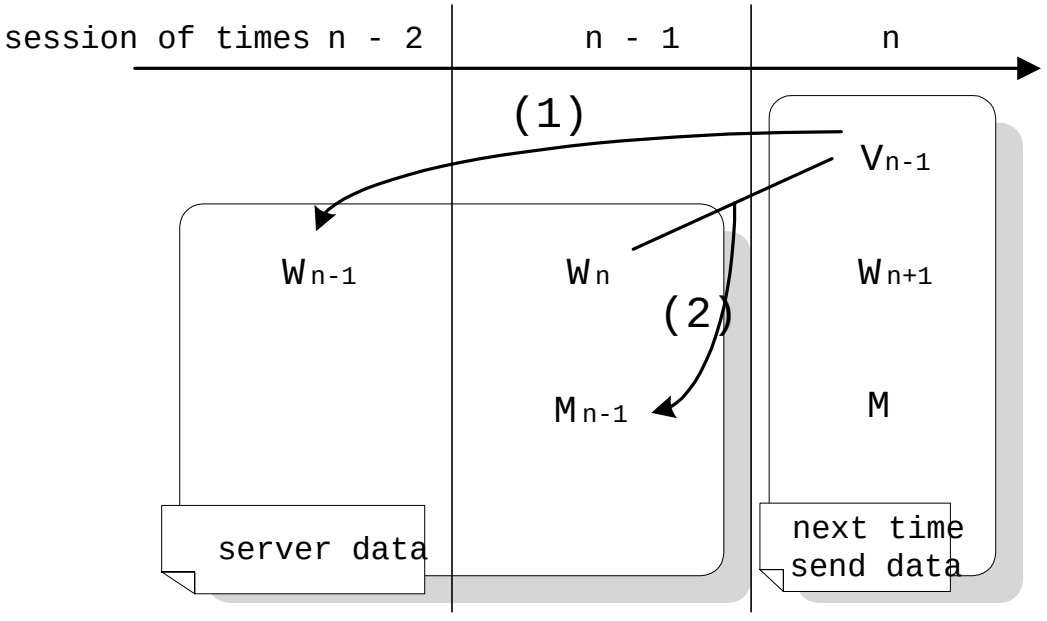

図 2.2 n 回目の PERM 認証

2.3 問題点

[1] [2] によるサービス方式は，認証に PERM 認証方式を用いている． PERM 認証方式は伝送経路上で 2 回連続して認証情報を取得する事により，第 3 者が正規のユーザとして成りすます可能がある．

また，このサービス方式では対象が電子メールに限られており，ネットワーク上でクライアントとサーバ間でファイル転送を行なうのに，外部のネットワークから自分のアカウントを有する内部ネットワークのサーバへアクセスしようとした場合，通常ファイアウォールによってそのアクセスを拒否される．例え，通過を許可していたとしても，ファイアウォールにセキュリティホールを作る事になり，ネットワーク全体のセキュリティを低下させるこ事になる．

第 3 章

コンテンツ転送サービス方式の提案

そこで， SAS-K (SAS - KUT version) 認証方式 [7] を用いた暗号化電子メール転送サービス方式をプラットホームに，出先の利用端末，内部サーバ，中継サーバの 3 者間による電子メール転送，ファイル転送を可能とするサービス方式を提案する．また，机上検討を行ない本方式の有効性を示す．

暗号化電子メール転送サービス方式は，中継サーバ - 出先の利用端末間ではクライアント / サーバシステムである．しかし，ファイル転送を前述と同じクライアント / サーバシステムで行なう場合，自分のアカウントを有する内部ネットワーク内のファイルをすべて中継サーバに蓄積しておき，同じ環境を構築する必要がある．それでは，あまりにもサーバに負荷がかかり過ぎ，現実的ではない．そこで，ユーザがファイル操作の要求を発信した時に，中継サーバを経由して，内部サーバへファイル操作を行なう方法が考えられる．

しかし，上記の方式においても，ユーザからのファイル操作要求，例えば，ファイルの転送要求，ファイル一覧の情報取得の要求など情報は，伝送系路上で盗聴され利用される恐れがある．ファイル操作コマンドの不正利用を防ぐ為には，ユーザからの要求情報に対し，SAS-K 認証に用いる認証子データを含ませ，暗号化することにより解決できると考える．

そこで，ファイル転送を行なう際の問題を解決する為に，SAS-K 認証方式を 3 者間で利用した．

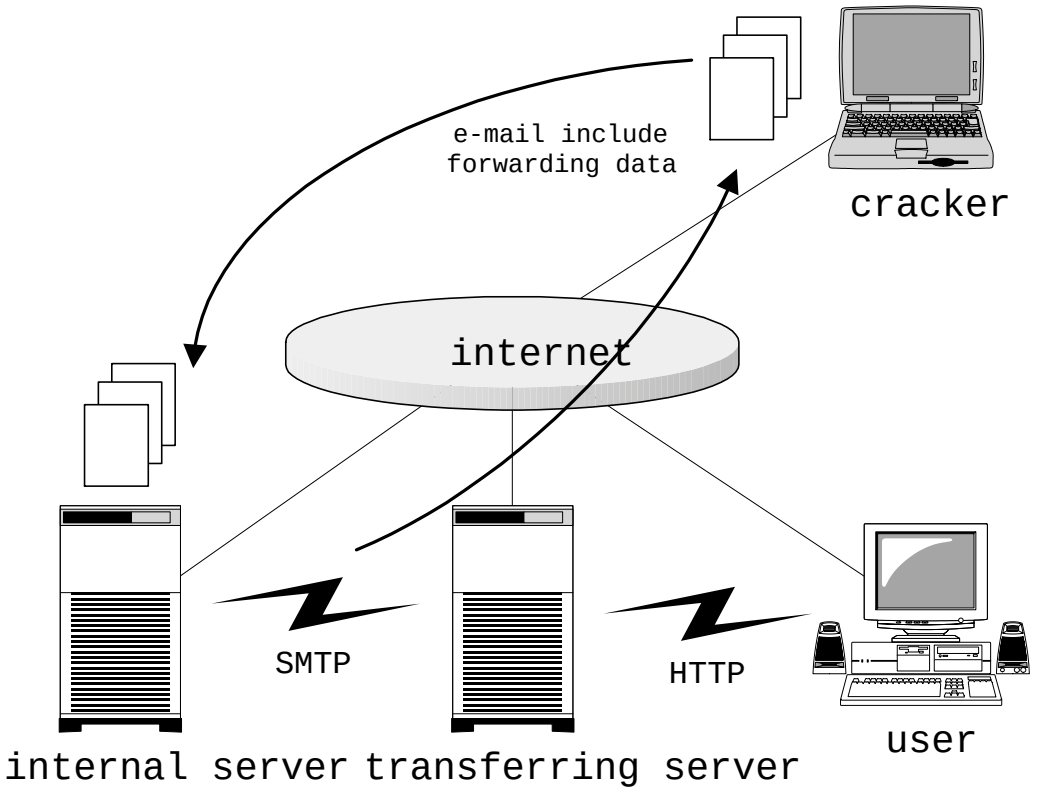

図 3.1　第 3 者によるなりすまし行為

3.1　3 者間での SAS-K 認証方式の利用

SAS-K 認証方式を利用した 3 者間 (出先の利用端末，内部サーバ，中継サーバ) での認証方式を提案する.

3.1.1　記号の定義

$A : ID$

$S : password$

$N_n : n$ 回目の認証で使用する乱数

E_n^1 : N_n を用いて 1 回，一方向性変換されたデータ

E_n^2 : N_n を用いて 2 回，一方向性変換されたデータ

E_n^3 : N_n を用いて 3 回，一方向性変換されたデータ

3.1.2　アルゴリズム

SAS-K 認証方式を利用した 3 者間での認証処理手順を説明する．図 3.2 に 3 者間による SAS-K 認証方式の流れを示す.

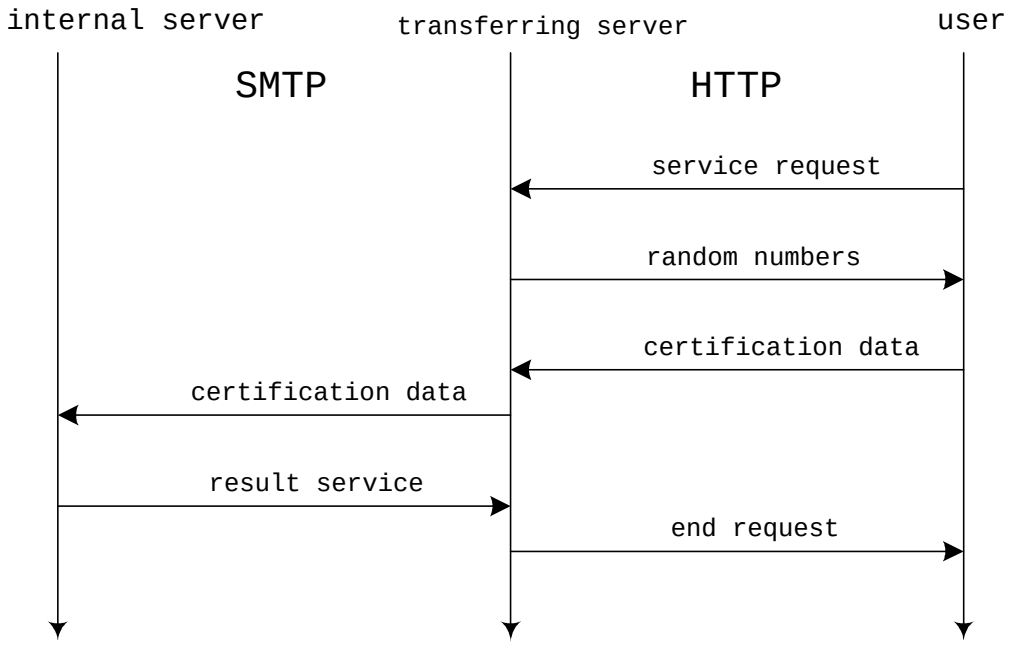

図 3.2　3 者間による SAS-K 認証方式の流れ

1. ユーザから中継サーバへ， A ，認証要求を送信する．
2. 認証要求を受け取った中継サーバは乱数 N_{n-1} を送信する．
3. 乱数 N_{n-1} を受け取ったユーザは，新たに乱数 N_n を生成し，

   $E^1_{n-1} \leftarrow E(A, S \oplus N_{n-1})$

   $E^2_{n-1} \leftarrow E(A, E^1_{n-1})$

   $E^1_n \leftarrow E(A, S \oplus N_n)$

   $E^2_n \leftarrow E(A, E^1_n)$

   $E^3_n \leftarrow E(A, E^2_n)$

   を算出する．
4. 算出されたデータを元に，$E^1_{n-1} \oplus E^2_{n-1} \oplus E^3_n$ ，$E^2_n \oplus E^2_{n-1}$ を生成し，A ，N_n と

一緒に中継サーバへ送信する．

5. 中継サーバは受け取った $E_{n-1}^1 \oplus E_{n-1}^2 \oplus E_n^3$, $E_n^2 \oplus E_{n-1}^2$ を使い，予め登録しておいた E_{n-1}^2 とで比較する．
6. 正当な認証情報であれば更に内部サーバへ $E_{n-1}^1 \oplus E_{n-1}^2 \oplus E_n^3$, $E_n^2 \oplus E_{n-1}^2$ ，A を送信する．
7. 内部サーバは受け取った $E_{n-1}^1 \oplus E_{n-1}^2 \oplus E_n^3$, $E_n^2 \oplus E_{n-1}^2$ を使い，予め登録しておいた E_{n-1}^2 とで比較する．
8. 正当な認証情報であれば，登録してある E_{n-1}^2 を E_n^2 に置き換え，中継サーバへ認証完了の情報を送信する．
9. 中継サーバが認証完了の情報を受け取ると，登録してある E_{n-1}^2 を E_n^2 に置き換え，ユーザへ認証完了の情報を送信する．

3.2　3 者間ファイル転送方式

出先の利用端末からファイル操作の要求を発信した時，自分のアカウントを有する内部ネットワーク内にあるプログラムを起動させファイル操作する方式にする．以下に，3 者間によるファイル転送の機能設計，及び，アルゴリズムを示す．

3.2.1　機能設計

ファイル転送に必要な機能を，各 3 者毎に以下のように設計する．

利用端末の機能

- ファイルの転送
- ファイルの一覧リストの表示
- 現在のワーキングディレクトリの表示
- ワーキングディレクトリの変更
- ファイルの復号

- ファイル操作の終了

ファイルの転送機能は，ファイル名を指定してファイル転送を行なう．ファイルの一覧リストの表示機能は，自分が作業しているディレクトリにあるファイルすべてのリストを表示する．現在のワーキングディレクトリの表示機能は，自分が作業しているディレクトリを表示する機能である．ワーキングディレクトリの変更機能は，自分が作業しているディレクトリを移動する．ファイルの復号機能は，ファイル転送したファイルを復号する．ファイル操作の終了機能は，ファイルの操作を終了する．

中継サーバの機能

- ユーザの認証
- ファイル操作の要求情報を送信
- ファイルの保管
- 認証情報，要求情報の暗号化

ユーザの認証機能は，ユーザが正当であるかどうかを判断する．ファイル操作の要求情報を送信機能は，利用端末から送られてきたファイル操作要求情報を，内部サーバへ送信する．ファイルの保管機能は，内部サーバから送信されてきたファイルを蓄積する．認証情報，要求情報の暗号化機能は，内部サーバへ送信する認証情報，要求情報を暗号化する．

内部サーバの機能

- ユーザの認証
- ファイル操作の要求を実行
- 認証情報，要求情報の復号
- ファイル操作の実行結果を暗号化
- ファイル操作の実行結果を送信

ユーザの認証機能は，送られてきた要求情報が正当なユーザから送信された情報か判断する．ファイル操作の要求を実行機能は，利用端末から発信されたファイル操作を実行する．

認証情報，要求情報の復号機能は，暗号化された情報を復号する．ファイル操作の実行結果を暗号化機能は，ファイル操作した結果を暗号化する．ファイル操作の実行結果を送信機能は，ファイル操作した結果を送信する．

3.2.2　アルゴリズム

3 者間によるファイル操作の処理は以下の流れになる (図 3.3)．利用端末 (図 3.4)，中継サーバ (図 3.5)，内部サーバ (図 3.6) のファイル操作のフローチャートを示す．

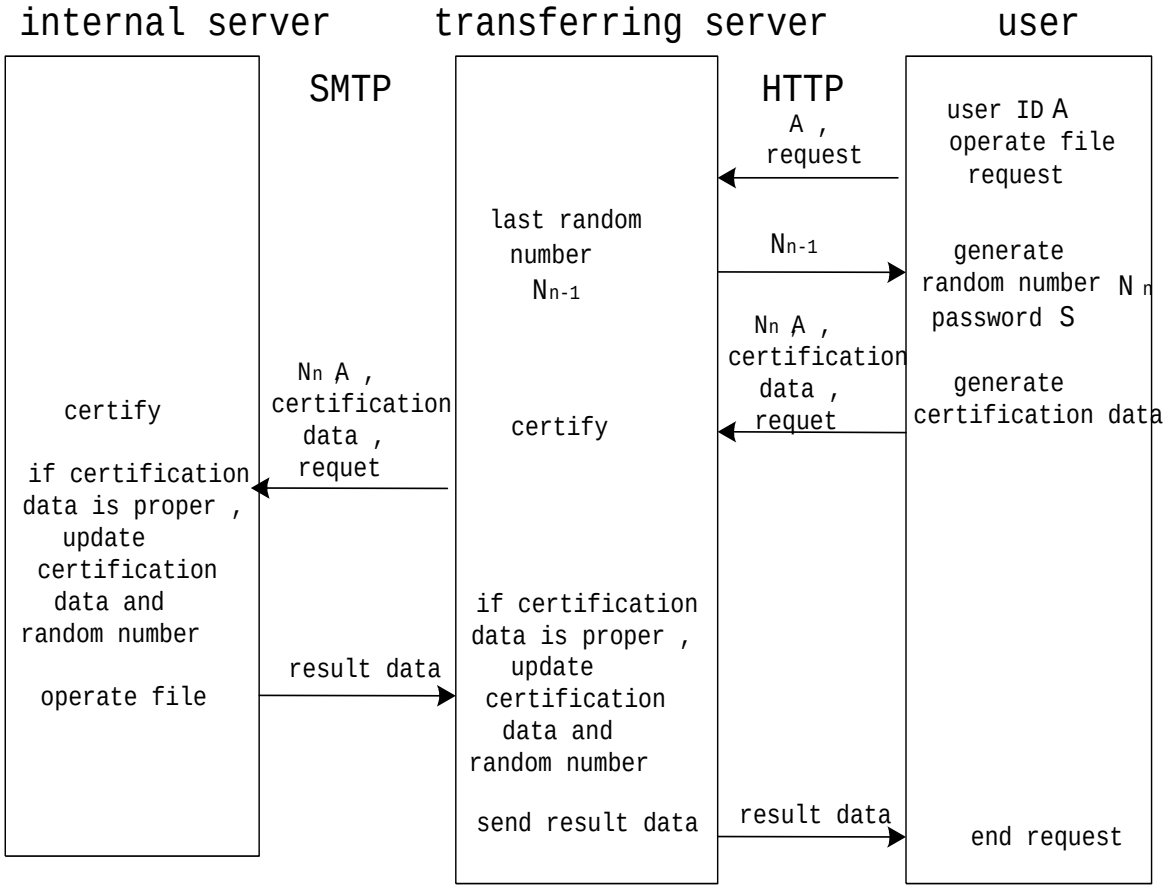

図 3.3　3 者間によるファイル操作の流れ

1. ユーザは ID とパスワードを入力し，中継サーバへファイル操作要求と ID を送信する．

2.　中継サーバは受け取った ID に対応して，登録してある認証に使用した前回の乱数をユーザへ送信する．

3.　ユーザは受け取った前回の乱数，新しく生成した乱数，入力した ID ，パスワードを使用して認証情報を生成する．

4.　ユーザは認証情報，新しく生成した乱数，ID を中継サーバへ送信する．

5.　中継サーバは受け取った認証情報，受け取った乱数，登録してある乱数を使用し認証を行なう．

6.　正当なユーザであると判断した場合，認証情報，ファイル操作要求，ユーザ側で新しく生成された乱数，ID を暗号化する．

7.　中継サーバは暗号化した情報を内部サーバへ送信する．

8.　内部サーバは暗号化された情報を受け取り，復号する．

9.　復号した情報から認証情報，ファイル操作要求，ユーザ側で新しく生成された乱数，ID を抽出し，抽出した認証情報，ユーザ側で新しく生成された乱数，登録してある乱数を使用して認証を行なう．

10.　正当なユーザであると判断した場合，ファイル操作要求に従って，ファイル操作を実行する．その際に，受け取った認証情報，ユーザ側で新しく生成された乱数を新たに登録する．

11.　ファイル操作を実行した結果を暗号化する．

12.　内部サーバは暗号化した情報を中継サーバへ送信する．

13.　中継サーバは暗号化された情報を受け取り，ユーザへ送信する．その際に，ユーザから受け取った認証情報，ユーザ側で新しく生成された乱数を新たに登録する．

14.　ユーザは暗号化された情報を受け取り，復号する．

3.2　3 者間ファイル転送方式

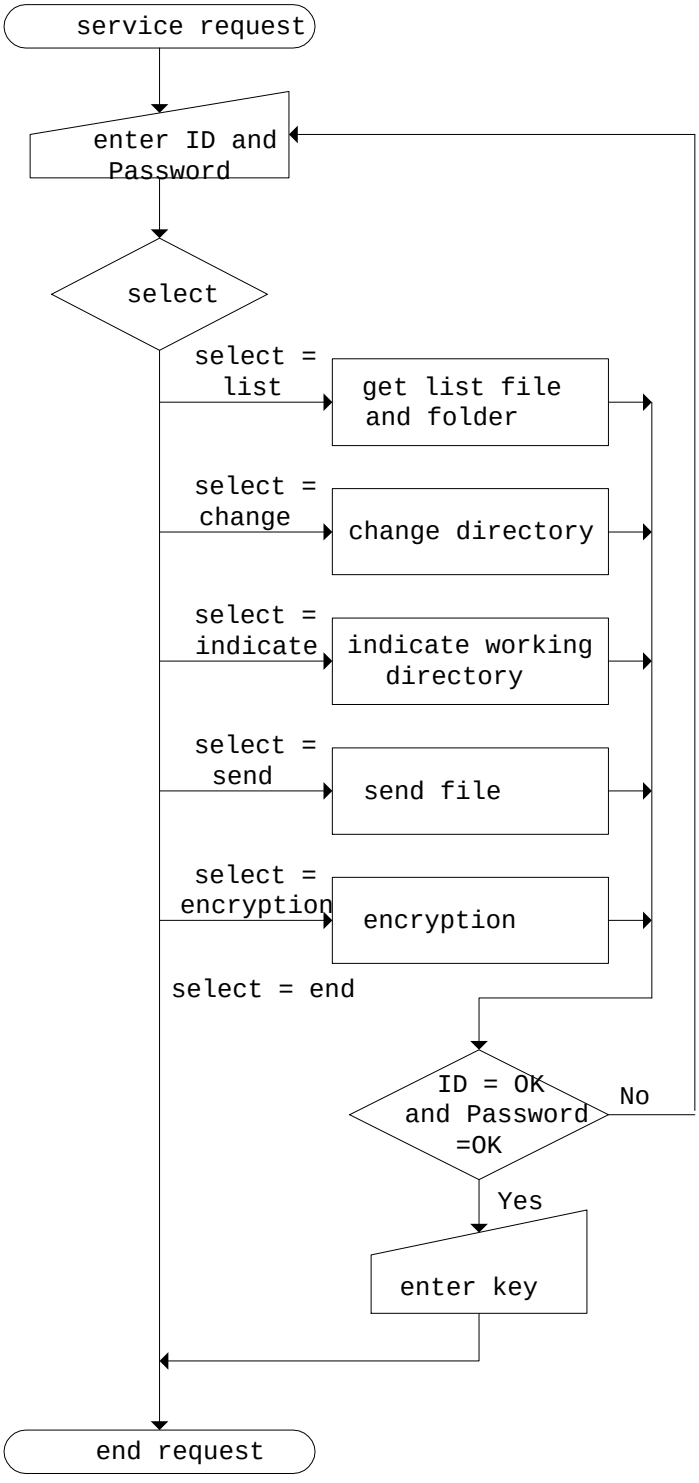

図 3.4　利用端末のファイル操作のフローチャート

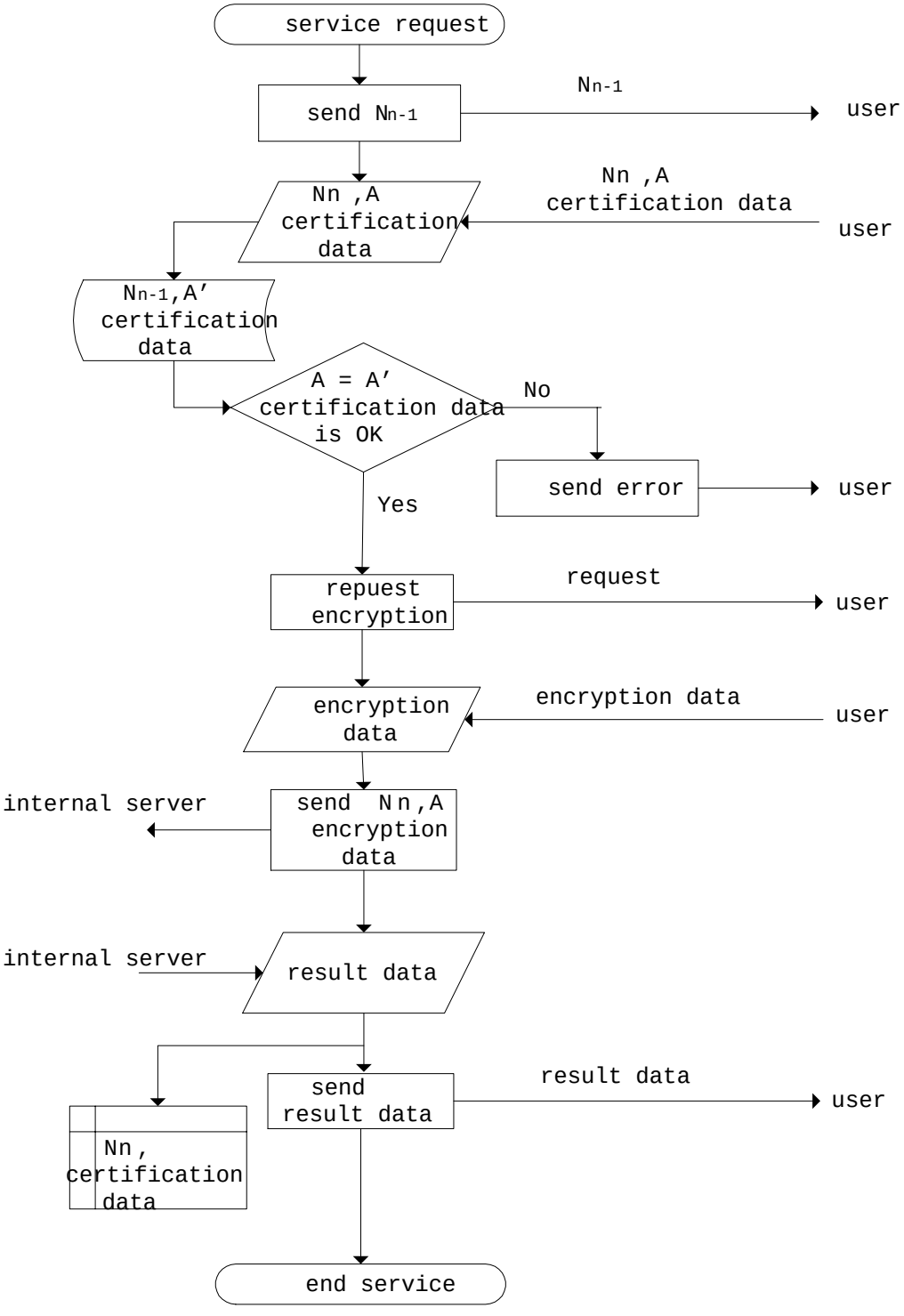

図 3.5　中継サーバのファイル操作のフローチャート

3.2　3 者間ファイル転送方式

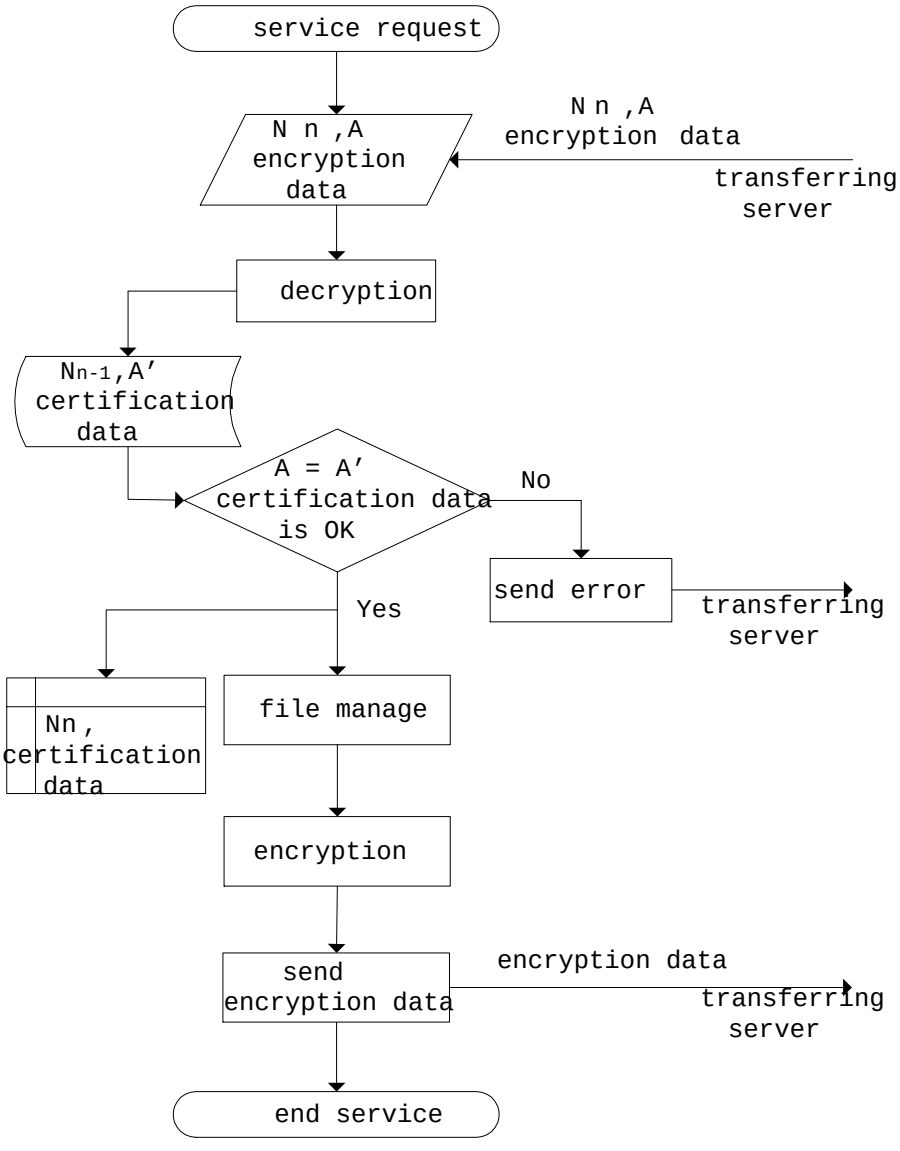

図 3.6　内部サーバのファイル操作のフローチャート

第 4 章

考察

第 3 章で提案した 3 者間での SAS-K 認証方式の利用とファイル転送方式に対して，ファイルの漏えい防止，ユーザの利便性について詳細な検討を行なう．

4.1 3 者間での SAS-K 認証方式の利用

SAS-K 認証方式，PERM 認証方式は共にワンタイムパスワードを用いた認証方式である．ワンタイムパスワードは伝送経路を流れる認証情報が随時更新されるため，第 3 者によるなりすまし行為が困難である．しかし，PERM 認証方式は認証情報を 2 回続けて取得する事による，なりすまし行為の危険性があった．SAS-K 認証方式はなりすまし行為の危険性がまったくない．有効性は [7] に示されている．

3 者間 SAS-K 認証方式では，利用端末 - 中継サーバ間，中継サーバ - 内部サーバ間のそれぞれに認証情報が流れる．第 3 者による認証情報の取得が行なわれた場合について検討を行なう．

利用端末 - 中継サーバ間で認証情報を取得し，第 3 者 - 中継サーバ間で使用した場合，認証情報は随時更新されるため，次回認証には使用できない．又，第 3 者 - 内部サーバ間で使用した場合，内部サーバへ送る認証情報は暗号化して送るため，取得した認証情報と一致しないので使用できない．

中継サーバ - 内部サーバ間で認証情報を取得し，第 3 者 - 中継サーバ間で使用した場合，取得した認証情報は暗号化されているため使用できない．又，第 3 者 - 内部サーバ間で使用した場合，認証情報は随時更新されるため，次回認証には使用できない．

4.2　ファイルの漏えい防止

本方式における秘密情報として SAS-K 認証で用いる認証パスワードと，ファイル操作を要求する時，ファイル操作実行の結果を送信する時に用いる暗号化パスワードがある．共に，パスワードそのものがネットワーク上を直接流れることはない．

中継サーバにおいても，第 3 者が攻撃した場合，中継サーバ自体にファイルが蓄積されていないため，ファイルが漏えいすることはない．この方式を，ファイルだけに限らず，従来の 3 者間電子メール転送サービス方式にも適応することにより，電子メールの漏えいも防ぐことができる．

また，ユーザからのファイル操作要求は，伝送経路上などで盗聴され再利用される恐れがあるため，送信する情報は要求情報に認証情報を加え暗号化したものを送信する．これにより，第 3 者によるなりすまし行為，いたずらを不可能にする．

4.3　ユーザの利便性

中継サーバを介することにより，不特定の端末からファイルの転送が可能となる．暗号化に関しても，共通鍵暗号方式を用い，ユーザが予め設定した暗号鍵で復号するので，復号鍵を記憶する必要がない．

第 5 章

まとめ

本研究では，従来の 3 者間電子メール転送サービス方式を利用した，3 者間ファイル転送方式を考案し，ユーザの要求毎にファイル操作を行なうアルゴリズム，機能設計を行なった．また， SAS-K 認証方式を適用することにより，第 3 者によるなりすまし行為を不可能とする，極めて安全性の高いサービス方式を提案した．

ユーザの利便性向上のため，ユーザインタフェースを設計し，3 者間ファイル転送方式の実装を行ない，併せて，データベースへアクセスを可能にする方式についても，検討して行く必要がある．

謝辞

本研究を行うに際し，多大なるご指導，ご鞭撻を頂いた，高知工科大学工学部情報システム工学科の清水　明宏助教授に深く感謝いたします.

また，本研究室院生　井上　富幸氏，岡田　実氏，田鍋　潤一郎氏，ならびに本研究室学部生　伊藤　雄君，大石　恭裕君，小橋　誠治君，篠原　直之君，竹内　紀貴君，谷藤　喜彦君，林　竜也君，　間城　昌厚君，安岡　隆司君，山崎　愛さんに深く感謝いたします.

参考文献

[1] 堀岡，戸田，清水，”電子メール転送サービス方式の検討”信学技報，TECHNICAL REPORT OF IEICE，OFS97-39，IE97-77 (1997-09)

[2] 堀岡，戸田，清水，”暗号化機能を有する電子メール転送サービス方式”信学技報，TECHNICAL REPORT OF IEICE，OFS97-42 (1997-11)

[3] 清水，宮口，”高速データ暗号アルゴリズム FEAL”信学論 D-I, Vol. J73-D-I, No. 7, pp. 630-636, Jul. 1987.

[4] 板原，白石，森井，清水，”アルゴリズム生成型 FEAL の強度評価に関する考察”信学技報，TECHNICAL REPORT OF IEICE, Vol. 100 Num. 101 pp.29-34 (2000.05)

[5] 岡本龍明，山本博資 著:「現代暗号」，産業図書株式会社，pp87-104 (1997 初版)

[6] 清水，”公衆電子メール転送サービスの検討”信学技報， Vol. 96, No. 380 pp. 18-24, Nov. 1996.

[7] 大石，林，井上，清水，”強力なパスワード認証方式の提案 (SAS-K)”投稿予定

付録 A

高速データ暗号アルゴリズム FEAL

FEAL は DES と同等の安全性を有し，8 / 16 ビットのマイクロプロセッサ上のソフトウェアで実現した場合，小さいプログラム規模で高速に処理できることを条件に開発された暗号方式であり， 64 ビットの共通鍵を用いて，64 ビットの平文を暗号化し， 64 ビットの暗号文を生成する方式である．

そして FEAL には，FEAL-NX (128 ビットの鍵を用いる) と FEAL-N (N は 4,8,16,32,… などがあり，暗号化に用いる基本変換単位の段数を表す) の 2 種類がある．

A.1 記号の定義

K : 暗号鍵 64 bit

K_i : i 段目の f 関数への鍵入力 16 bit

L_i : i 段目の平文上位 32 bit

R_i : i 段目の平文下位 32 bit

C_i : i 段目の鍵 64 bit の上位 32 bit

D_i : i 段目の鍵 64 bit の下位 32 bit

A.2 FEAL の基本構造

FEAL の基本構造について説明する．暗号化の処理の流れを図 A.1 に示す．

まず，前処理として $(K_{N+1}, K_{N+2}, K_{N+3}, K_{N+4})$ の鍵系列 64 ビットと，平文 64 ビットとの排他的論理和をとり，排他的論理を取ったビット列の上位 32 ビットと，下位 32 ビットをそれぞれ L'_0, R'_0 とし，さらに，$L_0 = L'_0, R_0 = R'_0 \oplus L'_0$ とおく．

次に，f 関数による変換を n 回繰り返す．n 回目の変換では，L_{n-1}, R_{n-1} と鍵 K_{n-1} を用いて，f 関数により L_n, R_n を出力する．

最後に，後処理として $L'_N = R_N \oplus L_N$ とし，R_N を上位 32 ビット，L'_N を下位 32 ビットとする 64 ビットのビット列を構成し，$(K_{N+5}, K_{N+6}, K_{N+7}, K_{N+8})$ の鍵系列 64 ビットと構成した 64 ビットのビット列との排他的論理和を取る．

また，FEAL の f 関数は図 A.2 のような構造をもつ．図 A.2 の $S_l(l = 0, 1)$ 関数は次の式で与えられる．

$$S_l = LRots2(x + y + l) mod 256$$

で，$LRot2$ は 2 ビットの左巡回シフトである．

鍵スケジューリングでは，暗号鍵 64 ビットの上位 32 ビットの C_i と，下位 32 ビットの D_i をひとまとまりとして，図 A.3 の変換を繰り返す．

また，鍵スケジューリングにおける f_K 関数は図 A.4 で与えられる．なお，図の鍵生成部一段当たり，2 個の鍵 K_{2n-1}, K_{2n} が生成されるので，鍵生成部の段数は，$(N + 8)/2$ 段必要となる．

A.2　FEAL の基本構造

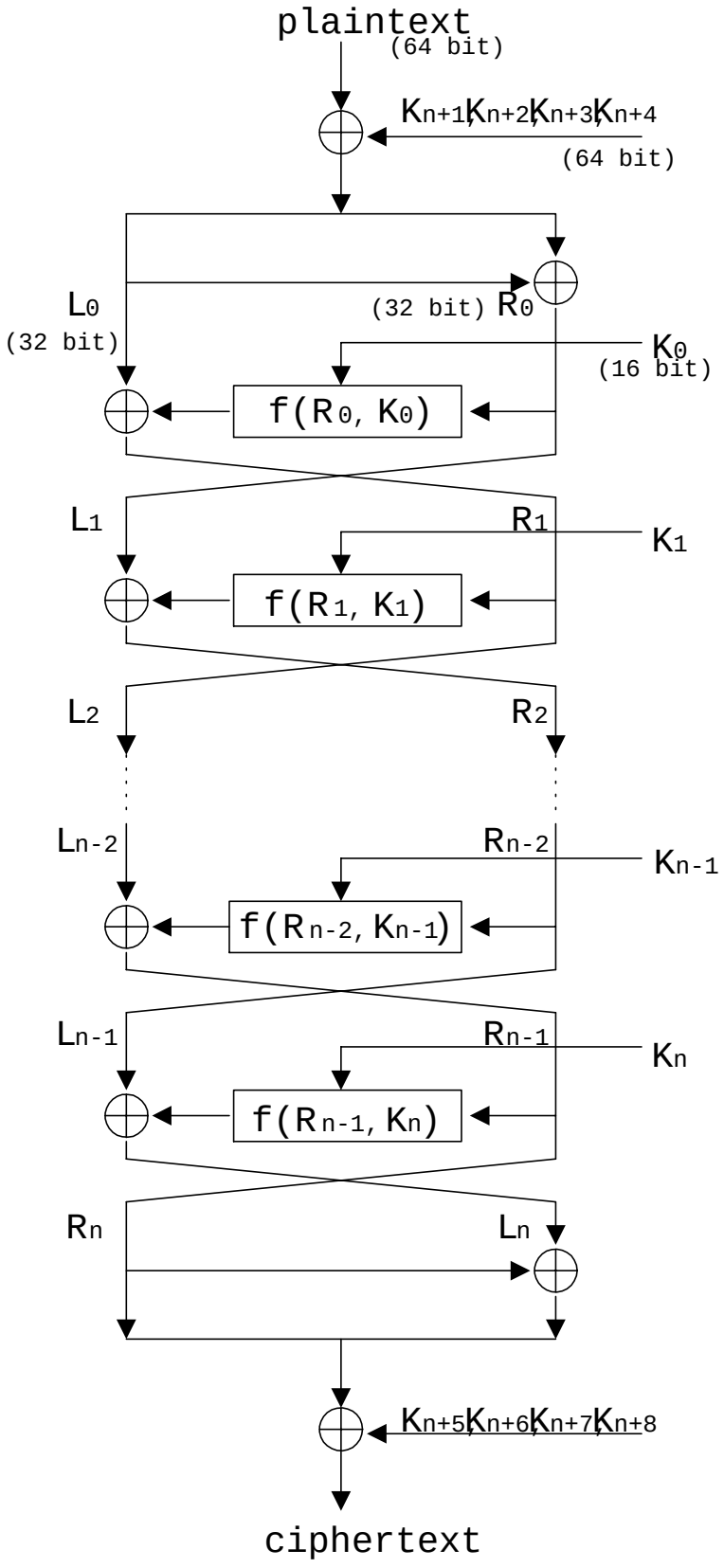

図 A.1　FEAL-N の暗号化の処理の流れ

A.2　FEAL の基本構造

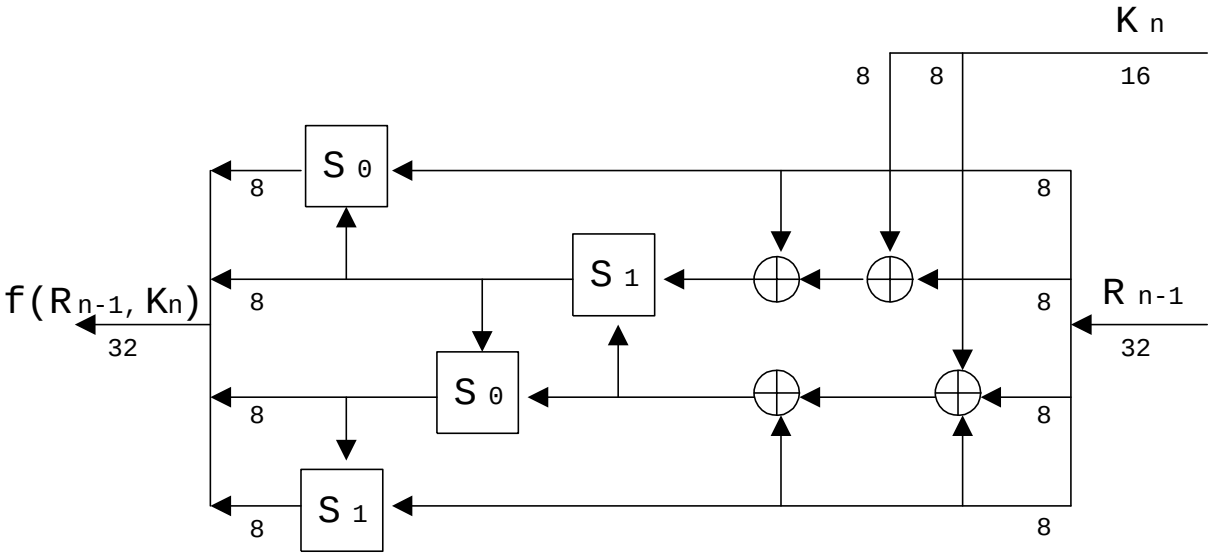

図 A.2　f 関数

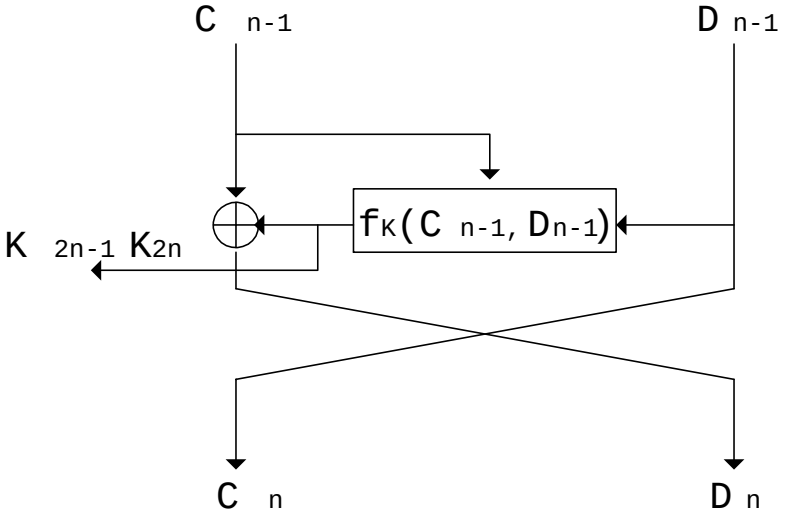

図 A.3　FEAL における鍵生成における基本単位

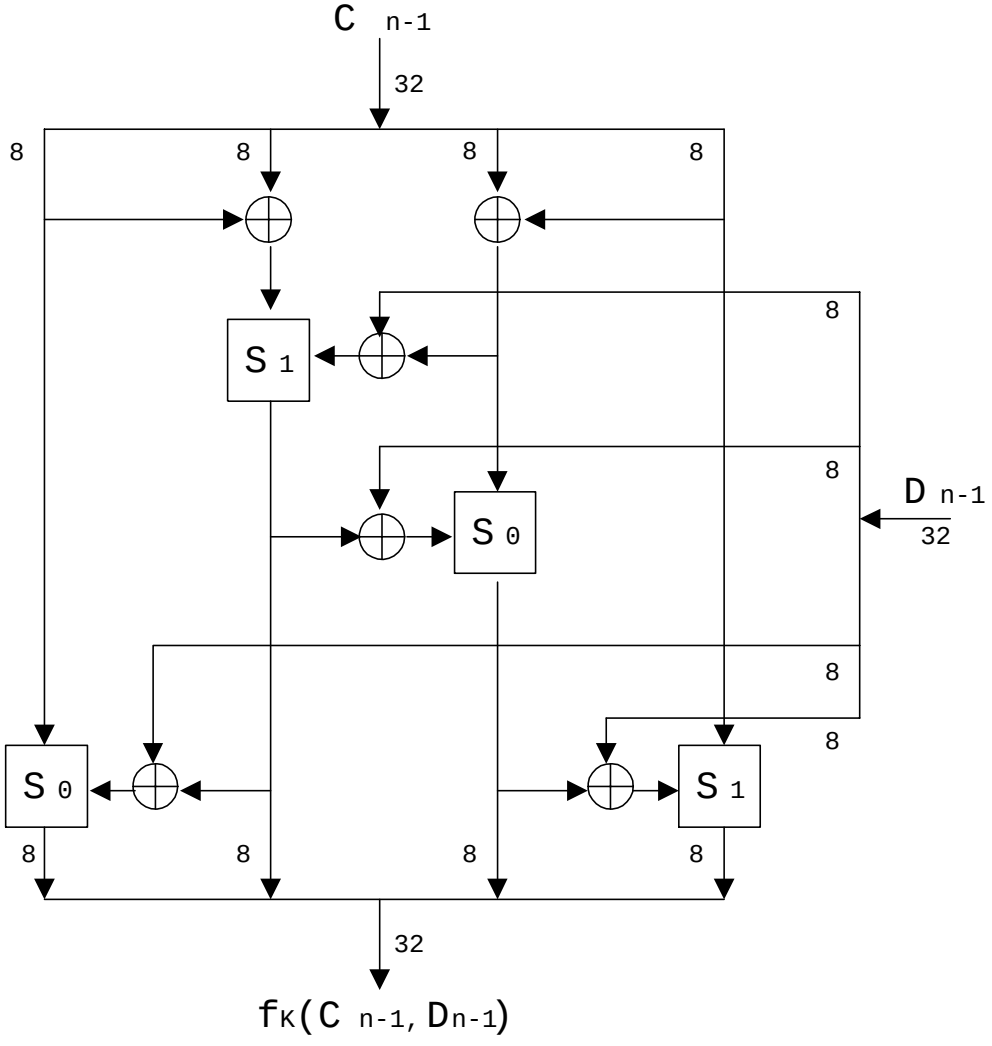

図 A.4　FEAL の鍵生成用 f_K 関数

付録 B

SAS-K（SAS - KUT version）認証方式

SAS-K 認証方式はワンタイムパスワード認証方式で，伝送路を流れる認証情報は毎回異なっているため，盗聴に強い安全な認証方式である．また，被認証者側および認証者側共に，実行する計算量が極めて少なく，被認証者側にも認証者側にも簡易で小さいプログラムサイズで実現可能.

B.1　記号の定義

$User$：　認証用プロトコルを用いるコンピュータユーザ

$Host$：　ユーザを認証するサーバ

A：　ユーザの ID

S：　ユーザのパスワード

n：　認証セッション回数を表す 0 以上の整数

N_n：　n 回目の認証に対応する乱数

E：　暗号用のハッシュ関数

E_n^m：　N_n を使い，$(S \oplus N_n)$ が m 回ハッシュされたことを示す

$\oplus$：　ビット毎の排他的論理和

B.2 SAS-K プロトコル

SAS-K プロトコルは登録フェーズと認証フェーズからなる．登録は一度だけ行なわれ，認証は $User$ がログインするたびに毎回行なわれる．その過程を図 B.1 に示す.

B.2.1 登録フェーズ $(n = 0)$

$User$： $E_0^2 = E^2(S \oplus N_0)$ を計算する.

$User$： A, E_0^2 を $Host$ に安全なチャンネルを用いて送信する.

$Host$： A, E_0^2 を保存する.

B.2.2 登録フェーズ $(n = k)$

$User$ に以下のデータを計算し，ユーザ ID A と共に $Host$ に送信する.

$A, E_{k-1}^1 \oplus E_{k-1}^2 \oplus E_k^3, E_k^2 \oplus E_{k-1}^2$

その後，$Host$ 側では，$(E_k^2 \oplus E_{k-1}^2) \oplus E_{k-1}^2 = E_k^2$ により，次回認証用データを取得する.

そして，$E(E_k^2) = E_k^3$ という認証確認用データを生成する.

次に，その E_k^3 を用いて，

$(E_{k-1}^1 \oplus E_{k-1}^2 \oplus E_k^3) \oplus E_k^3 = E_{k-1}^1 \oplus E_{k-1}^2$ を計算し，

さらに，登録されてある E_{k-1}^2 を用いて，

$(E_{k-1}^1 \oplus E_{k-1}^2) \oplus E_{k-1}^2 = E_{k-1}^1$ を導出する.

そしてそのデータにもう一度ハッシュ関数を適用し，E_{k-1}^2 を得る.

そしてこのデータと，登録されている E_{k-1}^2 を比較することによって認証を行なう．以上により，認証用データの正当性が検証された認証を可能とする.

B.2 SAS-K プロトコル

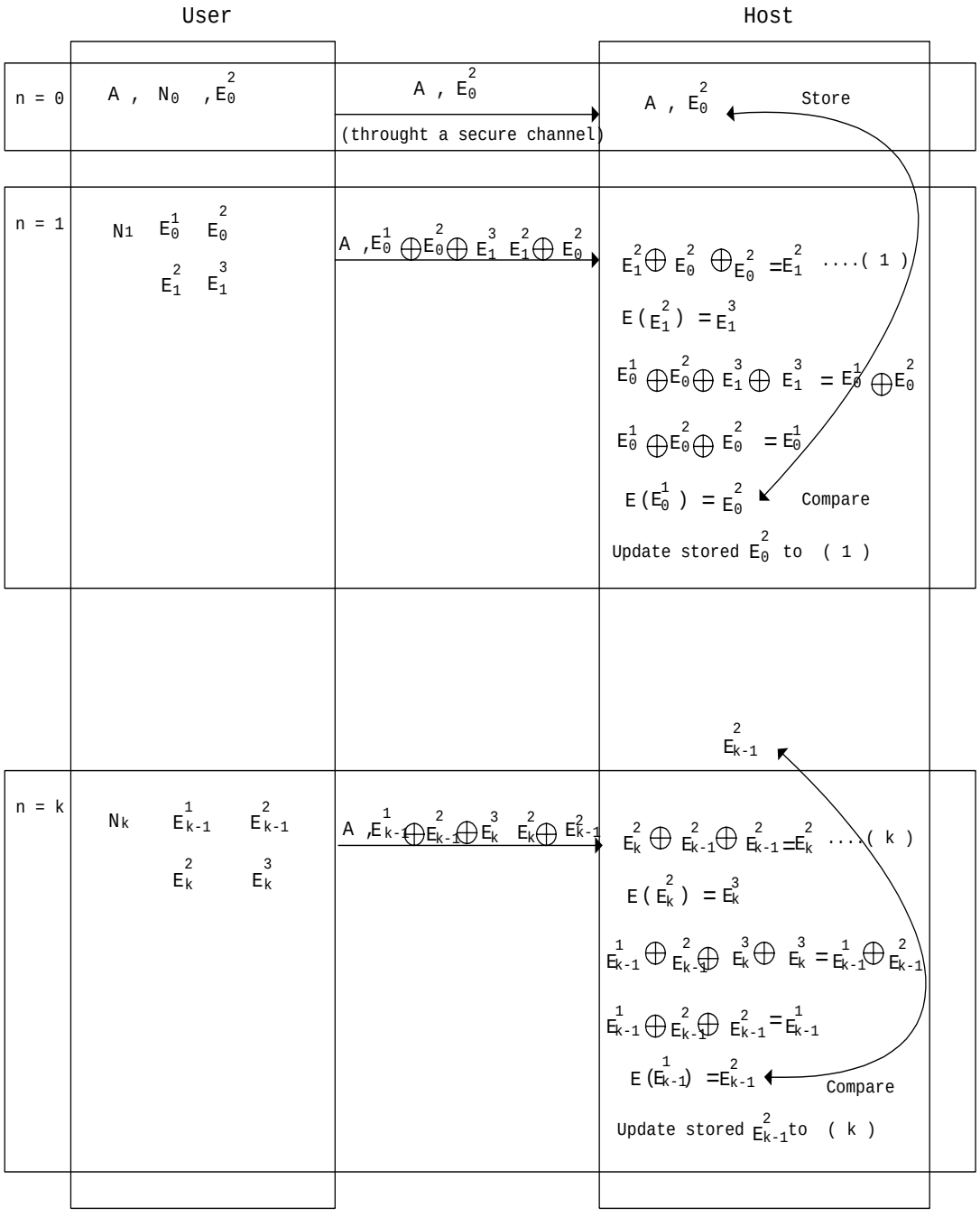

図 B.1 SAS-K の登録，及び認証フェーズ

B.2.3　同期対策

通信が切断される時として，$User$ から $Host$ への認証用データが未達の時，$Host$ から $User$ への認証 OK のメッセージが未達の時の 2 つが考えられる．この問題に対し，$User$, $Host$ 共にカウンタ (0，もしくは 1 の値を取る) を持ち，カウンタにより同期を取る (図 B.2).

登録フェーズで，$User$, $Host$ のカウンタをそろえておく (例えば 0 に)．認証フェーズで $User$ は，ユーザ ID ，認証用データに加え，カウンタの値も $Host$ に送信する．$Host$ 側では受け取ったカウンタの値を見て，同期が取れているかどうか検知．同期が取れていない場合は $User$ に対してエラーを返す．$User$ はメッセージを受け取り，認証 OK のメッセージが受け取れなかったことに気付く．そして新しい乱数を生成し，正常な認証を行なうことができる．

B.2　SAS-K プロトコル

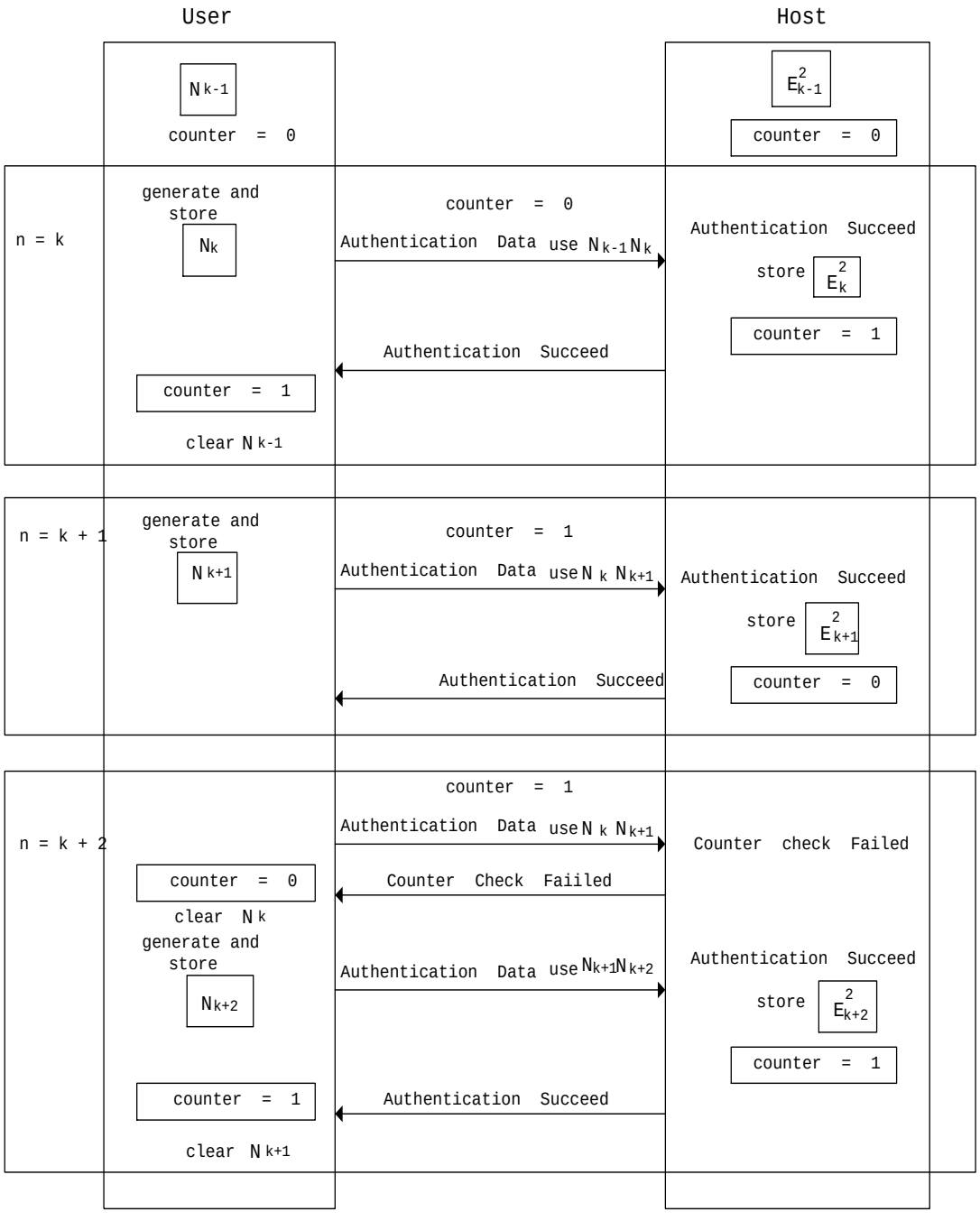

図 B.2　カウンタを利用した同期方法