

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2007-60568

(P2007-60568A)

(43) 公開日 平成19年3月8日(2007.3.8)

|                      |                 |             |
|----------------------|-----------------|-------------|
| (51) Int. Cl.        | F I             | テーマコード (参考) |
| HO4L 9/32 (2006.01)  | HO4L 9/00 675A  | 5B285       |
| GO6F 21/20 (2006.01) | HO4L 9/00 675D  | 5J104       |
|                      | GO6F 15/00 330B |             |

審査請求 未請求 請求項の数 18 O L (全 24 頁)

|           |                              |          |   |
|-----------|------------------------------|----------|---|
| (21) 出願番号 | 特願2005-246506 (P2005-246506) | (71) 出願人 | 300023383<br>株式会社トリニティーセキュリティーシステムズ<br>東京都品川区北品川1-20-9  |
| (22) 出願日  | 平成17年8月26日 (2005.8.26)       | (74) 代理人 | 100104190<br>弁理士 酒井 昭徳  |
|           |                              | (72) 発明者 | 清水 明宏<br>高知県高知市知寄町2丁目3番16-11<br>02号 アルファステイツ知寄町   |
|           |                              | (72) 発明者 | 辻 貴介<br>高知県香美郡土佐山田町旭町1-1-10<br>スバルK501  |
|           |                              | Fターム(参考) | 5B285 AA01 AA04 AA05 BA01 CA02<br>CA47 CB02 CB62 CB72 CB83<br>CB92 DA04<br>5J104 AA07 KA02 KA03 MA01 PA07 |

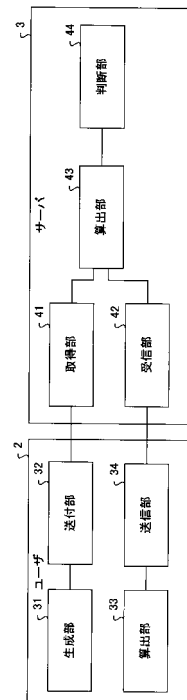
(54) 【発明の名称】 認証処理方法、認証処理プログラム、記録媒体および認証処理装置

(57) 【要約】

【課題】 認証処理の安全性を向上させること。

【解決手段】 ユーザ2を認証するサーバ3において、取得部41は、任意の値を用いて生成された今回の認証処理に用いる今回認証情報をユーザ2から取得する。受信部42は、次回の認証処理に用いる次回認証情報を今回認証情報を用いて隠蔽した第1の送信情報と、任意の値を次回認証情報を用いて隠蔽した第2の送信情報とを、ユーザ2から受信する。算出部43は、受信部42によって受信された第1の送信情報と取得部41によって取得された今回認証情報とを用いて次回認証情報を算出し、当該次回認証情報と第2の送信情報とを用いて任意の値を算出する。判断部44は、算出部43によって算出された任意の値と、取得部41によって取得された今回認証情報とに基づいて、ユーザ2を認証するか否かを判断する。

【選択図】 図3



## 【特許請求の範囲】

## 【請求項 1】

被認証装置を認証する認証装置における認証処理方法において、

任意の値を用いて生成された今回の認証処理に用いる今回認証情報を前記被認証装置から取得する取得工程と、

次回の認証処理に用いる次回認証情報を前記今回認証情報を用いて隠蔽した第 1 の送信情報と、前記任意の値を前記次回認証情報を用いて隠蔽した第 2 の送信情報とを、前記被認証装置から受信する受信工程と、

前記受信工程によって受信された前記第 1 の送信情報と前記取得工程によって取得された前記今回認証情報とを用いて前記次回認証情報を算出し、当該次回認証情報と前記第 2 10

の送信情報とを用いて前記任意の値を算出する算出工程と、  
前記算出工程によって算出された前記任意の値と、前記取得工程によって取得された前記今回認証情報とに基づいて、前記被認証装置を認証するか否かを判断する判断工程と、  
を含むことを特徴とする認証処理方法。

## 【請求項 2】

前記取得工程は、

前記任意の値に対して、演算前の値を算出することが困難な一方向変換関数による演算をおこなって生成された今回認証情報を取得し、

前記判断工程は、

前記任意の値に対して、前記一方向変換関数による演算をおこなった値が前記今回認証 20

情報と一致するか否かを判断することを特徴とする請求項 1 に記載の認証処理方法。

## 【請求項 3】

前記受信工程は、

第 1 の送信情報として、前記次回認証情報と前記今回認証情報とに対して、演算前の値を算出することが容易なマスク関数による演算をおこなった値を受信し、第 2 の送信情報として、前記任意の値と前記次回認証情報とに対して、前記マスク関数による演算をおこなった値を受信し、

前記算出工程は、

前記第 1 の送信情報と前記今回認証情報とに対して前記マスク関数による演算をおこなって前記次回認証情報を算出し、当該次回認証情報と前記第 2 の送信情報とに対して前記 30

マスク関数による演算をおこなって前記任意の値を算出することを特徴とする請求項 1 または 2 に記載の認証処理方法。

## 【請求項 4】

前記取得工程は、

前記今回認証情報とともに、前記被認証装置に固有の認証鍵を取得し、

前記受信工程は、

第 2 の送信情報として、前記次回認証情報に前記認証鍵を用いた所定の演算をおこなって得られた値によって前記任意の値を隠蔽した値を受信し、

前記算出工程は、

前記認証鍵と前記次回認証情報と前記第 2 の送信情報とを用いて前記任意の値を算出す 40

ることを特徴とする請求項 1 ~ 3 のいずれか一つに記載の認証処理方法。

## 【請求項 5】

前記取得工程は、

前記今回認証情報とともに、前記被認証装置に固有の認証鍵を取得し、

前記受信工程は、

第 1 の送信情報として、前記次回認証情報に前記認証鍵を用いた所定の演算をおこなって得られた値を、前記今回認証情報を用いて隠蔽した値を受信し、

前記算出工程は、

前記認証鍵と前記第 1 の送信情報と前記今回認証情報とを用いて前記次回認証情報を算 50

出することを特徴とする請求項 1 ~ 3 のいずれか一つに記載の認証処理方法。

## 【請求項 6】

前記取得工程は、  
 前記今回認証情報とともに、前記次回認証情報を暗号鍵として前記任意の値を暗号化した暗号情報を取得し、  
 前記判断工程は、  
 前記次回認証情報を暗号鍵として前記暗号情報を復号し、前記被認証装置を認証する可否かを判断することを特徴とする請求項 1 ~ 5 のいずれか一つに記載の認証処理方法。

## 【請求項 7】

前記取得工程は、  
 前記任意の値に対して、演算前の値を算出することが困難な一方向変換関数による演算を 2 回おこなって生成された今回認証情報を取得し、  
 前記判断工程は、  
 前記任意の値に対して、前記一方向変換関数による演算を 2 回おこなった値が前記今回認証情報と一致するか否かを判断することを特徴とする請求項 6 に記載の認証処理方法。

## 【請求項 8】

認証装置に認証を要求する被認証装置における認証処理方法において、  
 任意の値を用いて今回の認証処理に用いる今回認証情報を生成する生成工程と、  
 前記生成工程によって生成された今回認証情報を前記認証装置に送付する送付工程と、  
 次回の認証処理に用いる次回認証情報を前記今回認証情報を用いて隠蔽した第 1 の送信情報と、前記任意の値を前記次回認証情報を用いて隠蔽した第 2 の送信情報とを算出する算出工程と、  
 前記算出工程によって算出された前記第 1 の送信情報と前記第 2 の送信情報とを、前記認証装置に送信する送信工程と、  
 を含むことを特徴とする認証処理方法。

## 【請求項 9】

前記生成工程は、  
 前記任意の値に対して、演算前の値を算出することが困難な一方向変換関数による演算をおこなうことによって前記今回認証情報を生成することを特徴とする請求項 8 に記載の認証処理方法。

## 【請求項 10】

前記算出工程は、  
 第 1 の送信情報として、前記次回認証情報と前記今回認証情報とに対して、演算前の値を算出することが容易なマスク関数による演算をおこなった値を算出し、第 2 の送信情報として、前記任意の値と前記次回認証情報とに対して、前記マスク関数による演算をおこなった値を算出することを特徴とする請求項 8 または 9 に記載の認証処理方法。

## 【請求項 11】

前記生成工程は、  
 前記今回認証情報とともに、前記被認証装置に固有の認証鍵を生成し、  
 前記送付工程は、  
 前記今回認証情報とともに、前記認証鍵を前記認証装置に送付し、  
 前記算出工程は、  
 第 2 の送信情報として、前記次回認証情報に前記認証鍵を用いた所定の演算をおこなって得られた値によって前記任意の値を隠蔽した値を算出することを特徴とする請求項 8 ~ 10 のいずれか一つに記載の認証処理方法。

## 【請求項 12】

前記生成工程は、  
 前記今回認証情報とともに、前記被認証装置に固有の認証鍵を生成し、  
 前記送付工程は、  
 前記今回認証情報とともに、前記認証鍵を前記認証装置に送付し、  
 前記算出工程は、

10

20

30

40

50

第1の送信情報として、前記次回認証情報に前記認証鍵を用いた所定の演算をおこなって得られた値を、前記今回認証情報を用いて隠蔽した値を算出することを特徴とする請求項8～10のいずれか一つに記載の認証処理方法。

【請求項13】

前記生成工程は、  
前記今回認証情報とともに、前記次回認証情報を暗号鍵として前記任意の値を暗号化した暗号情報を生成し、  
前記送付工程は、  
前記今回認証情報とともに、前記暗号情報を前記認証装置に送付することを特徴とする請求項8～12のいずれか一つに記載の認証処理方法。

10

【請求項14】

前記生成工程は、  
前記任意の値に対して、演算前の値を算出することが困難な一方向変換関数による演算を2回おこなって前記今回認証情報を生成することを特徴とする請求項13に記載の認証処理方法。

【請求項15】

請求項1～14のいずれか一つに記載の認証処理方法をコンピュータに実行させることを特徴とする認証処理プログラム。

【請求項16】

請求項15に記載の認証処理プログラムを記録したコンピュータに読み取り可能な記録媒体。

20

【請求項17】

任意の値を用いて生成された今回の認証処理に用いる今回認証情報を被認証装置から取得する取得手段と、

次回の認証処理に用いる次回認証情報を前記今回認証情報を用いて隠蔽した第1の送信情報と、前記次回認証情報を用いて前記任意の値を隠蔽した第2の送信情報とを、前記被認証装置から受信する受信手段と、

前記受信手段によって受信された前記第1の送信情報と前記取得手段によって取得された前記今回認証情報とを用いて前記次回認証情報を算出し、当該次回認証情報と前記第2の送信情報とを用いて前記任意の値を算出する算出手段と、

30

前記算出手段によって算出された前記任意の値と、前記取得手段によって取得された前記今回認証情報とに基づいて、前記被認証装置を認証するか否かを判断する判断手段と、  
を備えることを特徴とする認証処理装置。

【請求項18】

任意の値を用いて今回の認証処理に用いる今回認証情報を生成する生成手段と、  
前記算出手段によって算出された今回認証情報を自装置が認証を要求する他装置に送付する送付手段と、

次回の認証処理に用いる次回認証情報を前記今回認証情報を用いて隠蔽した第1の送信情報と、前記任意の値を前記次回認証情報を用いて隠蔽した第2の送信情報とを算出する算出手段と、

40

前記算出手段によって算出された前記第1の送信情報と前記第2の送信情報とを、前記他装置に送信する送信手段と、

を備えることを特徴とする認証処理装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、情報通信システムなどにおいて、通信相手やユーザの利用資格を認証する認証処理方法、認証処理プログラム、記録媒体および認証処理装置に関する。

【背景技術】

50

## 【0002】

従来、認証者（サーバ）が被認証者（ユーザ）を認証する際に、ユーザにパスワードの入力を要求し、入力されたパスワードの正当性をもってユーザを認証するパスワード認証方式が広く用いられている。また、パスワード認証方式においてパスワードによる認証の安全性を担保するため、認証ごとに使い捨てのパスワードを用いるワンタイムパスワード方式や、パスワードそのものではなくパスワードから生成した認証情報を用いて認証をおこなう方式が知られている。

## 【0003】

例えば、SAS-2 (Simple And Secure password authentication protocol Ver. 2) 認証方式は、パスワード認証方式の一例であり、サーバは以下のような手順によってユーザを認証する（例えば、下記非特許文献1参照。）。図10および図11は、SAS-2認証方式におけるユーザ認証の処理工程を示すフローチャートである。

10

## 【0004】

なお、以下の説明において用いる記号および演算は、「 $\rightarrow$ 」は右辺の左辺への代入、「 $S$ 」はユーザが秘密に保持しているパスワード、「 $ID$ 」はユーザ識別子、「 $XOR$ 」は排他的論理和演算、「 $n$ 」は認証回数、「 $N_n$ 」は乱数（ $n$ は1以上の整数で、乱数を識別するために用いる）、をそれぞれ示している。また、「 $F$ 」「 $H$ 」はパスワード $S$ を用いない一方向変換関数、「 $X$ 」はパスワード $S$ と乱数 $N_n$ を用いる一方向変換関数で、 $X_n = X(ID, S \oplus N_n)$ 、をそれぞれ示している。

20

## 【0005】

まず、ユーザは、認証を受けたいサーバにあらかじめ登録をおこなっておく（以下、この登録作業を「初期登録」という）。ユーザの初期登録処理の手順を図10を参照して説明する。図10は、従来技術によるユーザの初期登録処理の手順を示すフローチャートである。ユーザは、あらかじめユーザ識別子 $ID$ およびパスワード $S$ を保持している。

## 【0006】

はじめに、ユーザは、乱数 $N_1$ を生成し、保存する（ステップS1001）。そして、乱数 $N_1$ 、秘密に保持しているパスワード $S$ 、ユーザ識別子 $ID$ を用いて、下記式（1）に示す初回認証情報 $A_1$ を算出し（ステップS1002）、ユーザ識別子 $ID$ とともに安全な手段でサーバに送付する（ステップS1003）。安全な手段とは、当該情報の専用線による送信や、当該情報を記憶した記憶媒体の郵送による送付などである。初回認証情報 $A_1$ は、初回（ $n=1$ ）の認証に用いる認証情報である。

30

$$A_1 = X_1(ID, S \oplus N_1) \cdots (1)$$

## 【0007】

サーバは、ステップS1003でユーザから送付されたユーザ識別子 $ID$ と、初回認証情報 $A_1$ とを対応づけて保存しておく（ステップS1004）。以上がユーザの初期登録作業の手順である。

## 【0008】

つぎに、初回（ $n=1$ ）以降、 $n$ 回目の認証時の認証処理について、図11を参照して説明する。図11は、初回（ $n=1$ ）以降、 $n$ 回目の認証時の認証処理の手順を示すフローチャートである。このとき、ユーザは、 $ID$ 、 $S$ 、 $N_n$ を保存している。また、サーバは $ID$ 、 $A_n$ （初回認証時は $n=1$ ）を保存している。まず、ユーザは保存している乱数 $N_n$ から、下記式（2）に示す $A_n$ を算出する（ステップS1101）。

40

$$A_n = X_n(ID, S \oplus N_n) \cdots (2)$$

## 【0009】

つぎに、さらに新しい乱数 $N_{n+1}$ を生成、保存するか、あるいは $A_n$ を $N_{n+1}$ として、 $N_{n+1}$ を保存する（ステップS1102）。そして、 $N_{n+1}$ を用いて、下記式（3）、（4）に示す $C$ 、 $D$ を算出し、算出した $C$ 、 $D$ 、および $A_n$ を用いて、下記式（5）、（6）に示す  $\rightarrow$  および  $\rightarrow$  を算出する（ステップS1103）。

$$C = X_n(ID, S \oplus N_{n+1}) \cdots (3)$$

50

$$D = F(ID, C), \dots (4)$$

$$C = XOR(D + A_n) \dots (5)$$

$$D = XOR(A_n) \dots (6)$$

## 【0010】

そして、算出した  $D$  を  $ID$  とともにサーバに送付する (ステップ S1104)。このとき、 $A_n$  は今回の認証処理に用いる今回認証情報、 $C$  は次回の認証処理に用いる次回認証情報、 $D$  は次回認証情報  $C$  を一方向性変換したもうひとつの次回認証情報である。

## 【0011】

サーバは、ユーザから  $A_n$  および  $C$  の送付を受けると、送付された  $A_n$  と  $C$  に対して、 $ID$  に対応して登録されている今回認証情報  $A_n$  を用いて、下記式 (7) に示す  $D$  を算出する。さらに、算出した  $D$  と今回認証情報  $A_n$  の和を用いて下記式 (8) に示す  $C$  を算出する (ステップ S1105)。

$$D = XOR(A_n) \dots (7)$$

$$C = XOR(D + A_n) \dots (8)$$

## 【0012】

つぎに、サーバは上記式 (8) によって算出した  $C$  を、 $ID$  とともに一方向性変換した結果と、 $D$  とが一致するかどうか ( $F(ID, C) = D?$ ) を検証する (ステップ S1106)。両者が一致する場合 (ステップ S1106: Yes)、ユーザに被認証者としての資格を認証し (認証成立)、次回 ( $n+1$  回目) の認証に用いる認証情報 ( $A_{n+1}$ ) として、次回認証情報  $C$  を保存する (ステップ S1107)。

## 【0013】

一方、両者が一致しない場合 (ステップ S1106: No)、認証不成立として (ステップ S1108)、本フローチャートによる処理を終了する。以上のような処理によって、サーバは、認証を求めてきたユーザに対して認証をおこなうか否かを判断する。

## 【0014】

【非特許文献1】辻貴介 (Takasuke TSUJI)、他2名、「シンプル アンド セキュア パスワード オーセンティケーション プロトコル バージョン2 (Simple And Secure password authentication protocol, Ver. 2 (SAS-2))」、電子情報通信学会技術研究報告書、2002年、OIS2002-30、Vol. 102、No. 314、p. 7-11

## 【発明の開示】

## 【発明が解決しようとする課題】

## 【0015】

しかしながら、上述した従来技術によれば、サーバ側に記録されている今回認証情報  $A$  を用いてマスク処理された送付情報を用いて認証処理をおこなう。このため、サーバ側に保存されている今回認証情報を盗むことによって、容易に送付情報を生成することが可能であり、悪意を持った第三者によって不正認証がおこなわれる可能性があるという問題点がある。

## 【0016】

特に、公共の場に設置されるサーバや、十分なセキュリティ知識を有していない者が設置したサーバなどは、悪意を有する者の標的にされやすく、今回認証情報が窃取されてしまう可能性が高い。また、サーバ側に悪意を有する者がいる場合、サーバに登録・保存されている今回認証情報を用いることによって、正当な被認証者に容易になりすまることができるという問題点がある。

## 【0017】

さらに、このように、なりすましなどによって不正な認証が成功すると、機密情報が漏洩したり、正当な被認証者情報が書き換えられてしまうおそれがある。一度公になってしまった情報を機密の状態に戻すことはできず、被認証者および認証者の双方に重大な損失が生じることとなるという問題点がある。

## 【0018】

10

20

30

40

50

本発明は、上述した従来技術による問題点を解消するため、認証処理の安全性をより向上させることができる認証処理方法、認証処理プログラム、記録媒体および認証処理装置を提供することを目的とする。

【課題を解決するための手段】

【0019】

上述した課題を解決し、目的を達成するため、請求項1の発明にかかる認証処理方法は、被認証装置を認証する認証装置における認証処理方法において、任意の値を用いて生成された今回の認証処理に用いる今回認証情報を前記被認証装置から取得する取得工程と、次回の認証処理に用いる次回認証情報を前記今回認証情報を用いて隠蔽した第1の送信情報と、前記任意の値を前記次回認証情報を用いて隠蔽した第2の送信情報とを、前記被認証装置から受信する受信工程と、前記受信工程によって受信された前記第1の送信情報と前記取得工程によって取得された前記今回認証情報とを用いて前記次回認証情報を算出し、当該次回認証情報と前記第2の送信情報とを用いて前記任意の値を算出する算出工程と、前記算出工程によって算出された前記任意の値と、前記取得工程によって取得された前記今回認証情報とに基づいて、前記被認証装置を認証するか否かを判断する判断工程と、を含むことを特徴とする。

10

【0020】

この請求項1の発明によれば、悪意を有する第三者が、認証装置による認証を不正に得ようとしても、今回認証情報の元となる任意の値を知ることができず、第2の送信情報を生成することができない。このため、認証装置が被認証装置から取得した今回認証情報が第三者によって窃取されたとしても、不正な認証がおこなわれることがない。

20

【0021】

また、請求項2の発明にかかる認証処理方法は、請求項1に記載の認証処理方法において、前記取得工程は、前記任意の値に対して、演算前の値を算出することが困難な一方向変換関数による演算をおこなって生成された今回認証情報を取得し、前記判断工程は、前記任意の値に対して、前記一方向変換関数による演算をおこなった値が前記今回認証情報と一致するか否かを判断することを特徴とする。

【0022】

この請求項2の発明によれば、演算前の値を算出することが困難な一方向変換関数による演算を、任意の値に対しておこなうことによって今回認証情報を生成する。このため、被認証装置からの取得過程や取得後の保存過程で、第三者によって今回認証情報が窃取されたとしても、今回認証情報を元にして任意の値を算出することはできず、第2の認証情報を生成することができない。

30

【0023】

また、請求項3の発明にかかる認証処理方法は、請求項1または2に記載の認証処理方法において、前記受信工程は、第1の送信情報として、前記次回認証情報と前記今回認証情報とに対して、演算前の値を算出することが容易なマスク関数による演算をおこなった値を受信し、第2の送信情報として、前記任意の値と前記次回認証情報とに対して、前記マスク関数による演算をおこなった値を受信し、前記算出工程は、前記第1の送信情報と前記今回認証情報とに対して前記マスク関数による演算をおこなって前記次回認証情報を算出し、当該次回認証情報と前記第2の送信情報とに対して前記マスク関数による演算をおこなって前記任意の値を算出することを特徴とする。

40

【0024】

この請求項3の発明によれば、演算前の値を算出することが容易なマスク関数による演算をおこなった値を受信する。このため、認証処理に用いられる今回認証情報や次回認証情報を、マスクした状態でやりとりすることができ、悪意を有する第三者によって送信情報を窃取されたとしても、不正な認証を防ぐことができる。

【0025】

また、請求項4の発明にかかる認証処理方法は、請求項1～3のいずれか一つに記載の認証処理方法において、前記取得工程は、前記今回認証情報とともに、前記被認証装置に

50

固有の認証鍵を取得し、前記受信工程は、第2の送信情報として、前記次回認証情報に前記認証鍵を用いた所定の演算をおこなって得られた値によって前記任意の値を隠蔽した値を受信し、前記算出工程は、前記認証鍵と前記次回認証情報と前記第2の送信情報とを用いて前記任意の値を算出することを特徴とする。

【0026】

この請求項4の発明によれば、被認証装置に固有の認証鍵を用いて算出した値を第2の送信情報として受信する。このため、万一固有の認証鍵を持たない第三者によって送信情報が取得されてしまったとしても、不正な認証を防止することができる。

【0027】

また、請求項5の発明にかかる認証処理方法は、請求項1～3のいずれか一つに記載の認証処理方法において、前記取得工程は、前記今回認証情報とともに、前記被認証装置に固有の認証鍵を取得し、前記受信工程は、第1の送信情報として、前記次回認証情報に前記認証鍵を用いた所定の演算をおこなって得られた値を、前記今回認証情報を用いて隠蔽した値を受信し、前記算出工程は、前記認証鍵と前記第1の送信情報と前記今回認証情報とを用いて前記次回認証情報を算出することを特徴とする。

10

【0028】

この請求項5の発明によれば、被認証装置に固有の認証鍵を用いて算出した値を第1の送信情報として受信する。このため、万一固有の認証鍵を持たない第三者によって送信情報が取得されてしまったとしても、不正な認証を防止することができる。

【0029】

また、請求項6の発明にかかる認証処理方法は、請求項1～5のいずれか一つに記載の認証処理方法において、前記取得工程は、前記今回認証情報とともに、前記次回認証情報を暗号鍵として前記任意の値を暗号化した暗号情報を取得し、前記判断工程は、前記次回認証情報を暗号鍵として前記暗号情報を復号し、前記被認証装置を認証するか否かを判断することを特徴とする。

20

【0030】

この請求項6の発明によれば、次回認証情報を暗号鍵として暗号化した暗号情報を取得する。これにより、取得工程によって取得した情報と受信工程によって取得した情報の出所が同一かを判断することができ、第三者による情報の偽造の有無を検出することができる。

30

【0031】

また、請求項7の発明にかかる認証処理方法は、請求項6に記載の認証処理方法において、前記取得工程は、前記任意の値に対して、演算前の値を算出することが困難な一方向変換関数による演算を2回おこなって生成された今回認証情報を取得し、前記判断工程は、前記任意の値に対して、前記一方向変換関数による演算を2回おこなった値が前記今回認証情報と一致するか否かを判断することを特徴とする。

【0032】

この請求項7の発明によれば、任意の値に対して一方向変換関数による演算を2回おこなった値によって正当な認証権限の有無を、同演算を1回おこなった値によって第三者による情報の偽造の有無を検出する。このように、それぞれの処理に用いる情報を異ならせることによって、不正な認証がおこなわれる可能性をより低くすることができる。

40

【0033】

また、請求項8の発明にかかる認証処理方法は、認証装置に認証を要求する被認証装置における認証処理方法において、任意の値を用いて今回の認証処理に用いる今回認証情報を生成する生成工程と、前記生成工程によって生成された今回認証情報を前記認証装置に送付する送付工程と、次の認証処理に用いる次回認証情報を前記今回認証情報を用いて隠蔽した第1の送信情報と、前記任意の値を前記次回認証情報を用いて隠蔽した第2の送信情報とを算出する算出工程と、前記算出工程によって算出された前記第1の送信情報と前記第2の送信情報とを、前記認証装置に送信する送信工程と、を含むことを特徴とする。

50



## 【0034】

この請求項8の発明によれば、悪意を有する第三者が、認証装置による認証を不正に得ようとしても、今回認証情報の元となる任意の値を知ることができず、第2の送信情報を生成することができない。このため、被認証装置から認証装置に送付した今回認証情報が第三者によって窃取されたとしても、不正な認証がおこなわれることがない。

## 【0035】

また、請求項9の発明にかかる認証処理方法は、請求項8に記載の認証処理方法において、前記生成工程は、前記任意の値に対して、演算前の値を算出することが困難な一方向変換関数による演算をおこなうことによって前記今回認証情報を生成することを特徴とする。

10

## 【0036】

この請求項9の発明によれば、演算前の値を算出することが困難な一方向変換関数による演算を、任意の値に対しておこなうことによって今回認証情報を生成する。このため、被認証装置からの送付過程や認証装置での保存過程で、第三者によって今回認証情報が窃取されたとしても、今回認証情報を元にして任意の値を算出することはできず、第2の認証情報を生成することができない。

## 【0037】

また、請求項10の発明にかかる認証処理方法は、請求項8または9に記載の認証処理方法において、前記算出工程は、第1の送信情報として、前記次回認証情報と前記今回認証情報とに対して、演算前の値を算出することが容易なマスク関数による演算をおこなった値を算出し、第2の送信情報として、前記任意の値と前記次回認証情報とに対して、前記マスク関数による演算をおこなった値を算出することを特徴とする。

20

## 【0038】

この請求項10の発明によれば、演算前の値を算出することが容易なマスク関数による演算をおこなった値を算出する。このため、認証処理に用いられる今回認証情報や次回認証情報を、マスクした状態でやりとりすることができ、悪意を有する第三者によって送信情報を窃取されたとしても、不正な認証を防ぐことができる。

## 【0039】

また、請求項11の発明にかかる認証処理方法は、請求項8～10のいずれか一つに記載の認証処理方法において、前記生成工程は、前記今回認証情報とともに、前記被認証装置に固有の認証鍵を生成し、前記送付工程は、前記今回認証情報とともに、前記認証鍵を前記認証装置に送付し、前記算出工程は、第2の送信情報として、前記次回認証情報に前記認証鍵を用いた所定の演算をおこなって得られた値によって前記任意の値を隠蔽した値を算出することを特徴とする。

30

## 【0040】

この請求項11の発明によれば、被認証装置に固有の認証鍵を用いて算出した値を第2の送信情報として算出する。このため、万一固有の認証鍵を持たない第三者によって送信情報が取得されてしまったとしても、不正な認証を防止することができる。

## 【0041】

また、請求項12の発明にかかる認証処理方法は、請求項8～10のいずれか一つに記載の認証処理方法において、前記生成工程は、前記今回認証情報とともに、前記被認証装置に固有の認証鍵を生成し、前記送付工程は、前記今回認証情報とともに、前記認証鍵を前記認証装置に送付し、前記算出工程は、第1の送信情報として、前記次回認証情報に前記認証鍵を用いた所定の演算をおこなって得られた値を、前記今回認証情報を用いて隠蔽した値を算出することを特徴とする。

40

## 【0042】

この請求項12の発明によれば、被認証装置に固有の認証鍵を用いて算出した値を第1の送信情報として算出する。このため、万一固有の認証鍵を持たない第三者によって送信情報が取得されてしまったとしても、不正な認証を防止することができる。

## 【0043】

50

また、請求項 13 の発明にかかる認証処理方法は、請求項 8 ~ 12 のいずれか一つに記載の認証処理方法において、前記生成工程は、前記今回認証情報とともに、前記次回認証情報を暗号鍵として前記任意の値を暗号化した暗号情報を生成し、前記送付工程は、前記今回認証情報とともに、前記暗号情報を前記認証装置に送付することを特徴とする。

【0044】

この請求項 13 の発明によれば、次回認証情報を暗号鍵として暗号化した暗号情報を今回認証情報とともに送付する。これにより、送付工程によって送付した情報と送信工程によって送信した情報の出所が同一であることを示すことができ、第三者による情報の偽造を防止することができる。

【0045】

また、請求項 14 の発明にかかる認証処理方法は、請求項 13 に記載の認証処理方法において、前記生成工程は、前記任意の値に対して、演算前の値を算出することが困難な一方向変換関数による演算を 2 回おこなって前記今回認証情報を生成することを特徴とする。

【0046】

この請求項 14 の発明によれば、任意の値に対して一方向変換関数による演算を 2 回おこなった値によって正当な認証権限を、同演算を 1 回おこなった値によって第三者による情報の偽造の有無を示すことができる。このように、それぞれの処理に用いる情報を異ならせることによって、不正な認証がおこなわれる可能性をより低くすることができる。

【0047】

また、請求項 15 の発明にかかる認証処理プログラムは、請求項 1 ~ 14 のいずれか一つに記載の認証処理方法をコンピュータに実行させることを特徴とする。

【0048】

この請求項 15 の発明によれば、請求項 1 ~ 14 のいずれか一つに記載の認証処理方法をコンピュータに実行させることができる。

【0049】

また、請求項 16 の発明にかかる記録媒体は、請求項 15 に記載の認証処理プログラムを記録したコンピュータに読み取り可能なことを特徴とする。

【0050】

また、請求項 16 の発明によれば、請求項 15 に記載の認証処理プログラムをコンピュータに読み取らせることができる。

【0051】

また、請求項 17 の発明にかかる認証処理装置は、任意の値を用いて生成された今回の認証処理に用いる今回認証情報を被認証装置から取得する取得手段と、次回の認証処理に用いる次回認証情報を前記今回認証情報を用いて隠蔽した第 1 の送信情報と、前記次回認証情報を用いて前記任意の値を隠蔽した第 2 の送信情報とを、前記被認証装置から受信する受信手段と、前記受信手段によって受信された前記第 1 の送信情報と前記取得手段によって取得された前記今回認証情報とを用いて前記次回認証情報を算出し、当該次回認証情報と前記第 2 の送信情報とを用いて前記任意の値を算出する算出手段と、前記算出手段によって算出された前記任意の値と、前記取得手段によって取得された前記今回認証情報とに基づいて、前記被認証装置を認証するか否かを判断する判断手段と、を備えることを特徴とする。

【0052】

この請求項 17 の発明によれば、悪意を有する第三者が、認証装置による認証を不正に得ようとしても、今回認証情報の元となる任意の値を知ることができず、第 2 の送信情報を生成することができない。このため、認証装置が被認証装置から取得した今回認証情報が第三者によって窃取されたとしても、不正な認証がおこなわれることがない。

【0053】

また、請求項 18 の発明にかかる認証処理装置は、任意の値を用いて今回の認証処理に用いる今回認証情報を生成する生成手段と、前記算出手段によって算出された今回認証情

10

20

30

40

50

報を自装置が認証を要求する他装置に送付する送付手段と、次回の認証処理に用いる次回認証情報を前記今回認証情報を用いて隠蔽した第1の送信情報と、前記任意の値を前記次回認証情報を用いて隠蔽した第2の送信情報とを算出する算出手段と、前記算出手段によって算出された前記第1の送信情報と前記第2の送信情報とを、前記他装置に送信する送信手段と、を備えることを特徴とする。

【0054】

この請求項18の発明によれば、悪意を有する第三者が、認証装置による認証を不正に得ようとしても、今回認証情報の元となる任意の値を知ることができず、第2の送信情報を生成することができない。このため、認証装置が被認証装置から取得した今回認証情報が第三者によって窃取されたとしても、不正な認証がおこなわれることがない。

10

【発明の効果】

【0055】

本発明にかかる認証処理方法、認証処理プログラム、記録媒体および認証処理装置によれば、認証処理の安全性をより向上させることができるという効果を奏する。

【発明を実施するための最良の形態】

【0056】

以下に添付図面を参照して、本発明にかかる認証処理方法、認証処理プログラム、記録媒体および認証処理装置の好適な実施の形態を詳細に説明する。

【0057】

(実施の形態1)

20

まず、実施の形態にかかる認証処理システムのシステム構成について、図1を参照して説明する。なお、以下に説明する実施の形態1~3において、認証処理システムのシステム構成、ハードウェア構成、機能的構成(図1~3に相当)は共通である。

【0058】

図1は、実施の形態にかかる認証処理システムのシステム構成を示す説明図である。認証処理システム1は、ユーザ2(2a~2f)、サーバ3によって構成される。認証処理システム1において、ユーザ2は認証者に認証を要求する被認証者、サーバ3は被認証者を認証する認証者である。また、ユーザ2a~2fおよびサーバ3は、ネットワーク4によって接続されている。

【0059】

30

ユーザ2は、サーバ3に認証を要求し、サーバ3が提供する所定のサービス(たとえばデータ通信接続サービス、ゲート通過許可、コンテンツ提供など)を受ける。サーバ3は、ユーザ2との間で認証処理をおこない、認証が成立した場合は、所定のサービスをおこなう。以下、サーバ3がユーザ2を認証した場合(認証が成立した場合は、ユーザ2との間で相互接続を開始するものとする。なお、詳細な処理の説明は省略するが、更なる安全性の向上のため、ユーザ2においても、認証要求先のサーバ3が正当な者であることを認証する相互認証をおこなってもよい。

【0060】

つぎに、認証処理システム1を構成するユーザ2、サーバ3のハードウェア構成について図2を参照して説明する。図2は、認証処理システムを構成するユーザ、サーバのハードウェア構成の一例を示すブロック図である。なお、以下では説明の便宜上、ユーザ2およびサーバ3は同一のハードウェア構成を有するものとする。また、ユーザ2、サーバ3の機能を実現するものを以下「装置」と呼ぶ。

40

【0061】

図2において、11は装置全体を制御するCPUを、12は基本入出力プログラムを記憶したROMを、13はCPU11のワークエリアとして使用されるRAMを、それぞれ示している。

【0062】

また、14はCPU11の制御にしたがってHD(ハードディスク)15に対するデータのリード/ライトを制御するHDD(ハードディスクドライブ)を、15はHDD14

50

の制御にしたがって書き込まれたデータを記憶するH Dを、それぞれ示している。また、16はCPU11の制御にしたがってF D（フレキシブルディスク）17に対するデータのリード/ライトを制御するF D D（フレキシブルディスクドライブ）を、17はF D D 16の制御にしたがって書き込まれたデータを記憶する着脱自在のF Dを、それぞれ示している。

【0063】

また、18はカーソル、メニュー、ウィンドウ、あるいは文字や画像などの各種データを表示するディスプレイを、19はネットワーク4への/からのデータの送受信をおこなうネットワークI/F（インターフェース）を、それぞれ示している。また、20は文字、数値、各種指示などの入力のための複数のキーを備えたキーボードを、21は各種指示の選択や実行、処理対象の選択、カーソルの移動などをおこなうマウスを、それぞれ示している。

10

【0064】

また、22は文字や画像を光学的に読み取るスキャナを、23は文字や画像を用紙上に印刷するプリンタを、24は着脱可能な記録媒体であるC D - R O Mを、25はC D - R O M 24に対するデータのリードを制御するC D - R O Mドライブを、26は上記各部を接続するためのバスまたはケーブルを、それぞれ示している。

【0065】

つぎに、実施の形態にかかる認証処理システム1の機能的構成について、図3を参照して説明する。図3は、認証処理システムを構成するユーザ、サーバの機能的構成を示すブロック図である。

20

【0066】

ユーザ2は、生成部31、送付部32、算出部33、送信部34によって構成される。生成部31は、任意の値（後述するa）を用いて今回の認証処理に用いる今回認証情報（後述するA）を生成する。生成部31は、具体的には、任意の値に対して、演算前の値を算出することが困難な一方向変換関数による演算をおこなうことによって今回認証情報を生成する。また、生成部31は、今回認証情報とともに、ユーザ2に固有の認証鍵（後述するK）を生成する。

【0067】

ここで、一方向変換関数とは、ある2つの値（x, yとする）をその関数によって演算した場合、その演算結果の値（zとする）と、ある2つの値のうち1つ（例えば、x）がわかっても、それらからもう1つの値（y）を算出することが困難な関数である。すなわち、一方向変換関数をhとしたとき、x, yをhによって演算した値をzとすると、zは、 $z = h(x, y)$ と表せる。このとき、x, zからyを算出することが困難な関数である。

30

【0068】

送付部32は、生成部31によって算出された今回認証情報をサーバ3に送付する。また、送付部32は、今回認証情報とともに、認証鍵をサーバ3に送付する。ここで、送付とは、当該情報の専用線による送信や、当該情報を記憶した記憶媒体の郵送による送付など、当該情報がサーバ3以外の者に知得されないような方法でサーバ3に送る、との意味である。

40

【0069】

算出部33は、次回の認証処理に用いる次回認証情報（後述するB）を算出し、今回認証情報を用いて次回認証情報を隠蔽した第1の送信情報と、次回認証情報を用いて任意の値を隠蔽した第2の送信情報とを算出する。算出部33は、具体的には、たとえば、第1の送信情報として、次回認証情報と今回認証情報とに対して、演算前の値を算出することが容易なマスク関数による演算をおこなった値を算出し、第2の送信情報として、次回認証情報（または次回認証情報および認証鍵の和）と任意の値とに対して、マスク関数による演算をおこなった値を算出する。

【0070】

50

ここで、マスク関数とは、2度同じ演算をすると、元の値が演算結果となる関数であり、例えば、排他的論理和演算 (XOR) が該当する。以下、マスク関数は排他的論理和演算として説明する。

【0071】

送信部34は、算出部33によって算出された第1の送信情報と第2の送信情報とを、サーバ3に送信する。送信部34は、たとえば、ネットワーク4を介して上述した送信情報をサーバ3に送信する。以上がユーザ2の機能的構成である。

【0072】

つづいて、サーバ3の機能的構成を説明する。サーバ3は、取得部41、受信部42、算出部43、判断部44によって構成される。取得部41は、ユーザ2の送付部32によって送付された今回認証情報およびユーザ2に固有の認証鍵を取得する。受信部42は、ユーザ2の送信部34によって送信された第1の送信情報と第2の送信情報とを受信する。

【0073】

算出部43は、受信部42によって受信された第1の送信情報と、取得部41によって取得された今回認証情報とを用いて次回認証情報を算出し、当該次回認証情報と第2の送信情報とを用いて任意の値を算出する。算出部43は、具体的には、たとえば、第1の送信情報と今回認証情報とに対して排他的論理和演算をおこなって次回認証情報(または次回認証情報および認証鍵の和)を算出し、当該次回認証情報(または次回認証情報および認証鍵の和)と第2の送信情報とに対して排他的論理和演算をおこなって任意の値を算出する。

【0074】

判断部44は、算出部43によって算出された任意の値と、取得部41によって取得された今回認証情報とに基づいて、ユーザ2を認証するか否かを判断する。判断部44は、具体的には、たとえば、任意の値に対して、一方向変換関数による演算をおこなった値が今回認証情報と一致するか否かを判断する。そして、一致する場合はユーザ2を認証し、一致しない場合はユーザ2の認証を拒否する。以上がサーバ3の機能的構成である。

【0075】

なお、上記各部は各装置のHD15やFD17、あるいはCD-ROM24などの各種記録媒体からRAM13に読み出されたプログラムの命令にしたがって、CPU11などが命令処理を実行することにより、それぞれの機能を実現する。

【0076】

(認証者 - 被認証者間の認証処理)

つぎに、サーバ3(認証者)が、ユーザ2(被認証者)を認証する際の処理について説明する。認証処理に先立って、サーバ3に対してユーザ2の初期登録処理がおこなわれる。そして、初期登録処理時にサーバ3に登録された情報を用いて、サーバ3がユーザ2を認証する認証処理がおこなわれる。

【0077】

なお、以下の説明において用いる記号および演算は、「 $\rightarrow$ 」は右辺の左辺への代入、「 $S$ 」はユーザ(被認証者)が秘密に保持しているパスワード、「 $ID$ 」はユーザ(被認証者)識別子、「 $XOR$ 」は排他的論理和、「 $n$ 」は認証回数、「 $N_n$ 」は乱数( $n$ は1以上の整数で、乱数を識別するために用いる)、「 $F$ 」はパスワード $S$ を用いない一方向変換関数であり、 $z = F(x, y)$ とすると、 $z$ と $x$ から $y$ を算出することが計算量的に困難な関数である。また、「 $X$ 」はパスワード $S$ と乱数 $N_n$ を用いる一方向変換関数で、 $X_n = X(ID, S, N_n)$ とする。

【0078】

まず、ユーザ2の初期登録処理について図4を参照して説明する。図4は、ユーザの初期登録処理の手順を示すフローチャートである。図4において、ユーザ2は、複数存在するユーザ2から自身を識別するためのユーザ識別子 $ID$ を保持している。また、ユーザ2は、自身のみが知る秘密のパスワード $S$ を保持している。ユーザ識別子 $ID$ およびパスワ

ードSは、ユーザ2に記録されていてもよいし、処理の都度入力を要求することとしてもよい。

【0079】

はじめに、ユーザ2は、秘密鍵Kおよび乱数 $N_1$ を生成し、保存する(ステップS101)。そして、ユーザ識別子ID、パスワードS、乱数 $N_1$ を用いて、下記式(9)、(10)の手順によって $A_1$ を算出する(ステップS102)。ここで、式(10)に示す $A_1$ は、初回( $n=1$ )の認証に用いる認証情報 $A_n$ である。以下、 $A_1$ を初回認証情報という。

$$a = X(ID, S, N_1) \cdots (9)$$

$$A_1 = F(ID, a) \cdots (10)$$

10

【0080】

つぎに、ユーザ2はサーバ3に対して、ユーザ識別子ID、秘密鍵K、初回認証情報 $A_1$ を、安全な手段によって送信する(ステップS103)。ここで、安全な手段とは、当該情報の専用線による送信や、当該情報を記憶した記憶媒体の郵送による送付などである。また、ユーザ2は、算出した初回認証情報 $A_1$ を保存し(ステップS104)、本フローチャートによる処理を終了する。

【0081】

サーバ3は、ユーザ2から送信された秘密鍵K、初回認証情報 $A_1$ を、それぞれユーザ識別子IDに関連づけて保存して(ステップS105)、本フローチャートによる処理を終了する。以上のような処理によって、ユーザ2がサーバ3に初期登録される。

20

【0082】

つぎに、初回( $n=1$ )以降、 $n$ 回目の認証時の認証処理について、図5を参照して説明する。図5は、初回( $n=1$ )以降、 $n$ 回目の認証処理の手順を示すフローチャートである。このとき、ユーザ2が保存している保存情報は、秘密鍵K、乱数 $N_n$ (初回認証の場合は $n=1$ )、認証情報 $A_n$ (初回認証の場合は $n=1$ )である。また、サーバ3が保存している保存情報は、ユーザ2のID、図4の初期登録時にユーザ2から送信された秘密鍵Kおよび認証情報 $A_n$ (初回認証の場合は $n=1$ )である。

【0083】

まず、ユーザ2は、保存している $N_n$ から、下記式(11)に示す $a$ を算出する(ステップS151)。また、さらに新しい乱数 $N_{n+1}$ を生成、保存し(ステップS152)、

30

下記式(12)、(13)の手順によってBを算出する(ステップS153)。

$$a = X(ID, S, N_n) \cdots (11)$$

$$b = X(ID, S, N_{n+1}) \cdots (12)$$

$$B = F(ID, b) \cdots (13)$$

【0084】

つぎに、ユーザ2は、算出した $a$ 、Bおよび保存しているK、 $A_n$ を用いて、下記式(14)、(15)に示す $B'$ を算出し(ステップS154)、ID、 $B'$ をサーバ3に送信する(ステップS155)。また、次回認証情報 $A_{n+1}$ として、ステップS153で算出したBを保存して(ステップS156)、ユーザ2は本フローチャートによる処理を終了する。また、 $a$ は認証情報 $A_n$ の元となるデータである。

40

$$B' = B \text{ XOR } A_n \cdots (14)$$

$$(B + K) \text{ XOR } a \cdots (15)$$

【0085】

サーバ3は、ユーザ2から受信したIDと $B'$ に対して、ユーザ2のIDに対応して登録されている認証情報 $A_n$ を用いて、下記式(16)に示すB、およびBとKとの和を用いて下記式(17)に示す $a$ を算出する(ステップS157)。

$$B = B' \text{ XOR } A_n \cdots (16)$$

$$a = B \text{ XOR } (B + K) \cdots (17)$$

【0086】

そして、算出した $a$ と保存しているIDとを一方向性変換した結果 $F(ID, a)$ がA

50

$A_n$ に等しいかどうかを判断し(ステップS158)、等しい場合は(ステップS158: Yes)、ユーザ2に対して被認証者の資格を認証し、ユーザ2との接続を開始する(ステップS159)。また、次回の認証に用いる認証情報( $A_{n+1}$ )としてBを保存して(ステップS160)、サーバ3は本フローチャートによる処理を終了する。一方、F(ID, a)が $A_n$ に等しくない場合は(ステップS158: No)、認証不成立として、ユーザ2にエラーメッセージを送信して(ステップS161)、サーバ3は本フローチャートによる処理を終了する。

#### 【0087】

なお、ステップS154における、の算出方法は、上記式(14)、(15)に示した演算に限られるものではない。上記式(14)、(15)に示した演算では、を算出する際にBにKを加算しているが、この処理は、がサーバ3への送信過程で第三者に取得されてしまった場合などに、取得された値を利用した攻撃を防ぐためのものである。すなわち、の算出に用いるB、または、の算出に用いるBのいずれか一方に演算を施し、とのB部分( $A_n$ あるいはaと排他的論理和演算する値)を一致させないようにして、からBが知られないようにしている。の算出方法は、このような条件を満たすものであれば、上記式(14)、(15)の演算には限られず、様々なバリエーションが存在する。

10

#### 【0088】

たとえば、IDなどのユーザ2とサーバ3との間の共有情報、あるいは $A_n$ などの共有情報から生成した値をKとして用いることが可能である。上述した例では、Kは図4のステップS101において秘密鍵として算出した値であるが、たとえば、ユーザ識別子IDを用いて、下記式(18)、(19)のように演算してもよい。

20

$$\begin{aligned} B \text{ XOR } A_n \cdots (18) \\ (B + ID) \text{ XOR } a \cdots (19) \end{aligned}$$

#### 【0089】

また、の算出にあたって、Bに対してKを加算するのではなく、下記式(20)、(21)に示すようにBからKを減算することとしてもよい。この他、乗算や除算など、BとKとの間でどのような演算をおこなってもよい。

$$\begin{aligned} B \text{ XOR } A_n \cdots (20) \\ (B - K) \text{ XOR } a \cdots (21) \end{aligned}$$

30

#### 【0090】

また、の算出においても、Kを用いた演算をおこなうこととしてもよい。たとえば、下記式(22)、(23)に示すように、の算出においてBにKを加算することとしてもよい。ただし、の算出におけるBへの演算は、B-Kのように、逆演算が可能なものでなくてはならない。

$$\begin{aligned} (B + K) \text{ XOR } A_n \cdots (22) \\ B \text{ XOR } a \cdots (23) \end{aligned}$$

#### 【0091】

以上のような算出方法のバリエーションに加え、下記式(24)、(25)のように、これらのバリエーションを組み合わせた演算によって、を算出することとしてもよい。

40

$$\begin{aligned} (B - ID) \text{ XOR } A_n \cdots (24) \\ B \text{ XOR } a \cdots (25) \end{aligned}$$

#### 【0092】

このように、の算出方法にはバリエーションがあるが、どのような演算によって、が算出されたかによって、ステップS157におけるaの算出方法が異なってくる。たとえば、上記式(18)、(19)に示した演算によって、を算出した場合、下記式(26)、(27)に示す演算によってaを算出することができる。

$$\begin{aligned} B \text{ XOR } A_n \cdots (26) \\ a \text{ XOR } (B + ID) \cdots (27) \end{aligned}$$

50

## 【0093】

以上説明したように、実施の形態1にかかる認証処理システムによれば、認証処理に用いる認証情報は、被認証者（ユーザ2）のみが保存する認証情報の元となるデータ（パスワード $S$ 、乱数 $N_n$ ）を用いなければ生成することができない。このため、認証者（サーバ3）に保存されている被認証者に関する情報（ $ID$ 、 $K$ 、 $A_n$ ）を第三者が盗んだとしても、第三者は認証処理に用いる認証情報を生成することができず、認証者による認証を受けることができない。

## 【0094】

（実施の形態2）

つづいて、実施の形態2にかかる認証処理システムについて説明する。実施の形態1にかかる認証処理システムでは、認証処理に用いる認証情報を、被認証者（ユーザ2）のみが保存するデータを元に生成することによって、認証者（サーバ3）からの被認証者情報の窃取に対応できるようにした。実施の形態2にかかる認証処理システムでは、さらに、次回（ $n+1$ 回目）の認証に用いる次回認証情報（ $A_{n+1} : B$ ）を用いて暗号化した情報から、今回（ $n$ 回目）の認証に用いる今回認証情報（ $A_n$ ）の元となる情報が算出されるかを確認する。これにより、第三者による送付情報の偽造を検出することができる。

## 【0095】

まず、ユーザ2の初期登録処理について図6を参照して説明する。図6は、ユーザの初期登録処理の手順を示すフローチャートである。図6において、ユーザ2は、複数存在するユーザ2から自身を識別するためのユーザ識別子 $ID$ を保持している。また、ユーザ2は、自身のみが知る秘密のパスワード $S$ を保持している。

## 【0096】

はじめに、ユーザ2は、秘密鍵 $K$ を生成し、保存する（ステップ $S201$ ）。また、乱数 $N_1$ 、 $N_2$ を生成し、 $N_2$ を保存する（ステップ $S202$ ）。そして、ユーザ識別子 $ID$ 、パスワード $S$ 、乱数 $N_1$ 、 $N_2$ を用いて、下記式（28）、（29）の手順によって $A$ を、下記式（30）～（32）の手順によって $a_1$ を算出する（ステップ $S203$ ）。ここで、 $A_1$ および $a_1$ は、初回（ $n=1$ ）の認証に用いる初回認証情報である。また、下記式（32）に示す $E_{\{B\}}(a)$ は、 $B$ を鍵として $a$ を暗号化した結果の値である。

$$a = X(ID, S, N_1) \cdots (28)$$

$$A_1 = F(ID, a) \cdots (29)$$

$$b = X(ID, S, N_2) \cdots (30)$$

$$B = F(ID, b) \cdots (31)$$

$$a_1 = E_{\{B\}}(a) \cdots (32)$$

## 【0097】

つぎに、ユーザ2はサーバ3に対して、ユーザ識別子 $ID$ 、秘密鍵 $K$ 、初回認証情報 $A_1$ 、 $a_1$ を、安全な手段によって送信する（ステップ $S204$ ）。また、算出した $a$ 、 $b$ （以下、認証子 $a$ 、 $b$ という）および $A_1$ 、 $B$ を保存して（ステップ $S205$ ）、ユーザ2は本フローチャートによる処理を終了する。

## 【0098】

サーバ3は、ユーザ2から送信された秘密鍵 $K$ 、初回認証情報 $A_1$ 、 $a_1$ を、それぞれユーザ識別子 $ID$ にそれぞれ関連づけて保存して（ステップ $S206$ ）、本フローチャートによる処理を終了する。以上のような処理によって、ユーザ2がサーバ3に初期登録される。

## 【0099】

つぎに、初回（ $n=1$ ）以降、 $n$ 回目の認証時の認証処理について、図7を参照して説明する。図7は、初回（ $n=1$ ）以降、 $n$ 回目の認証処理の手順を示すフローチャートである。このとき、ユーザ2が保存している情報は、秘密鍵 $K$ 、乱数 $N_{n+1}$ （初回認証の場合は $n=1$ のため $N_2$ ）、認証情報 $A_n$ （初回認証の場合は $n=1$ ）、 $B$ 、認証子 $a$ 、 $b$ である。サーバ3が保存している情報は、図6に示した初期登録時にユーザ2から送信された秘密鍵 $K$ および認証情報 $A_n$ 、 $a_n$ （初回認証の場合は $n=1$ ）である。

10

20

30

40

50



## 【0100】

まず、ユーザ2は、保存している  $N_{n+1}$  から、下記式(33)に示す  $b$  を算出し(ステップS251)、保存している  $b$  と等しいかを判断する(ステップS252)。このとき、 $b$  の算出に用いる  $ID$ 、 $S$  は、認証処理の都度ユーザ2の利用者に入力をもよおし、ユーザ2で保存しておいてもよい。保存している  $b$  と等しい場合は(ステップS252: Yes)、被認証者としての資格を認証し、ステップS253に移行する。一方、保存している  $b$  と等しくない場合は(ステップS252: No)、被認証者としての資格を認証せず、本フローチャートによる処理を終了する。

$$b = X(ID, S, N_{n+1}) \cdots (33)$$

## 【0101】

また、ユーザ2は、新しい乱数  $N_{n+2}$  を生成、保存し(ステップS253)、下記式(34)、(35)の手順によって認証子  $c$  および認証情報  $C$  を算出する(ステップS254)。このとき、認証子  $c$  および認証情報  $C$  は、次々回の認証に用いる次々回認証情報である。

$$c = X(ID, S, N_{n+2}) \cdots (34)$$

$$C = F(ID, c) \cdots (35)$$

## 【0102】

つぎに、ユーザ2は、算出した  $b$ 、 $C$  および保存している  $K$ 、 $A_n$ 、 $B$  を用いて、下記式(36)~(38)に示す  $a$ 、 $b_{n+1}$  を算出する(ステップS255)。そして、サーバ3に  $ID$ 、 $a$ 、 $b_{n+1}$  を送信する(ステップS256)。また、次回認証子  $b$ 、次回認証情報  $B$  として、算出した次々回認証子  $c$ 、次々回認証情報  $C$  を保存して(ステップS257)、ユーザ2は本フローチャートによる処理を終了する。なお、 $a$ 、 $b_{n+1}$  の算出方法は、実施の形態1で説明したように、さまざまなバリエーションが存在するが、以下では、その一例として下記式(36)、(37)に示す演算をおこなうものとする。

$$B \oplus A_n \cdots (36)$$

$$(B + K) \oplus a \cdots (37)$$

$$b_{n+1} = E_{\{C\}}(b) \cdots (38)$$

## 【0103】

サーバ3は、ユーザ2から受信した  $a$ 、 $b_{n+1}$  および  $ID$  に対応して保存されている認証情報  $A_n$ 、秘密鍵  $K$  を用いて、下記式(39)、(40)の手順によって  $a$  を算出する(ステップS258)。

$$B \oplus A_n \cdots (39)$$

$$a = (B + K) \oplus b_{n+1} \cdots (40)$$

## 【0104】

そして、サーバ3は、算出した  $a$  および保存している  $ID$  を一方向性変換した結果  $F(ID, a)$  が  $A_n$  に等しいかどうかを判断し(ステップS259)、等しい場合は(ステップS259: Yes)、ユーザ2に対して被認証者の資格を認証する。つづいて、サーバ3は、受信した  $B$  を用いて暗号化情報  $D_{\{B\}}(a)$  を復号し(  $D_{\{B\}}(a)$  と示す)、 $D_{\{B\}}(a)$  が  $a$  に等しいかを判断する(ステップS260)。

## 【0105】

$D_{\{B\}}(a)$  が  $a$  に等しい場合は(ステップS260: Yes)、認証情報  $B$  が改ざんされていないことが証明され、ユーザ2との接続を開始する(ステップS261)。また、次回( $n+1$ 回目)の認証処理時に用いる認証情報( $A_{n+1}$ )として  $B$  を保存して(ステップS262)、本フローチャートによる処理を終了する。

## 【0106】

一方、ステップS259において、 $F(ID, a)$  が  $A_n$  に等しくない場合は(ステップS259: No)、認証不成立として、ユーザ2にエラーメッセージを送信して(ステップS263)、本フローチャートによる処理を終了する。また、ステップS260において、 $D_{\{B\}}(a)$  が  $a$  に等しくない場合は(ステップS260: No)、認証情報  $B$  が改ざんされているとして、ユーザ2にエラーメッセージを送信して(ステップS2

10

20

30

40

50

63)、本フローチャートによる処理を終了する。

【0107】

以上説明したように、実施の形態2にかかる認証処理システムによれば、認証処理に用いる認証情報は、被認証者(ユーザ2)のみが保存する認証情報の元となるデータ(パスワードS、乱数 $N_{n+1}$ )を用いなければ生成することができない。このため、認証者(サーバ3)に保存されている被認証者に関する情報(ID, K,  $A_n, a_n$ )を第三者が盗んだとしても、第三者は認証処理に用いる認証情報を生成することができず、認証者による認証を受けることができない。

【0108】

さらに、次回認証情報(B)を用いて暗号化した情報( $a_n: E\{B\}(a)$ )から、今回の認証に用いる今回認証情報の元となるデータ( $a: D\{B\}(a_n)$ )が算出されるかを確認することによって、第三者による送信情報の偽造を検出することができる。

【0109】

(実施の形態3)

つづいて、実施の形態3にかかる認証処理システムについて説明する。実施の形態3にかかる認証処理システムでは、送付情報の偽造検出に用いるデータと、認証処理に用いるデータを異なるデータとする。これにより、認証処理時の安全性を向上させることができる。

【0110】

まず、ユーザ2の初期登録処理について説明する。図8は、ユーザの初期登録の処理手順を示すフローチャートである。図8において、ユーザ2は、複数存在するユーザ2から自身を識別するためのユーザ識別子IDを保持している。また、ユーザ2は、自身のみが知る秘密のパスワードSを保持している。

【0111】

はじめに、ユーザ2は、秘密鍵Kを生成し、保存する(ステップS301)。また、乱数 $N_1, N_2$ を生成し、 $N_2$ を保存する(ステップS302)。そして、ユーザ識別子ID、パスワードS、乱数 $N_1$ を用いて、下記式(41)~(43)に示す手順によって $A'_1$ を、下記式(44)~(47)の手順によって $a_1$ を算出する(ステップS303)。ここで、 $A'_1$ および $a_1$ は、初回( $n=1$ )の認証に用いる初回認証情報である。また、下記式(47)に示す $E\{B'\}(a)$ は、 $B'$ を鍵としてaを暗号化した結果の値である。

$$a = X(ID, S, N_1) \dots (41)$$

$$A = F(ID, a) \dots (42)$$

$$A'_1 = F(ID, A) \dots (43)$$

$$b = X(ID, S, N_2) \dots (44)$$

$$B = F(ID, b) \dots (45)$$

$$B' = F(ID, B) \dots (46)$$

$$a_1 = E\{B'\}(a) \dots (47)$$

【0112】

つぎに、ユーザ2はサーバ3に、ユーザ識別子ID、秘密鍵K、初回認証情報 $A'_1, a_1$ を安全な手段によって送信する(ステップS304)。また、ユーザ2は、算出した $a, b$ (以下、認証子 $a, b$ という)および $A, A'_1, B, B'$ を保存して(ステップS305)、本フローチャートによる処理を終了する。

【0113】

サーバ3は、ユーザ2から送信された秘密鍵K、初回認証情報 $A'_1, a_1$ を、それぞれユーザ識別子IDに関連づけて保存して(ステップS306)、本フローチャートによる処理を終了する。以上のような処理によって、ユーザ2がサーバ3に初期登録される。

【0114】

つぎに、初回( $n=1$ )以降、 $n$ 回目の認証時の認証処理について、図9を参照して説明する。図9は、初回( $n=1$ )以降、 $n$ 回目の認証処理の手順を示すフローチャートで

10

20

30

40

50

ある。このとき、ユーザ 2 が保存している情報は、秘密鍵  $K$ 、乱数  $N_{n+1}$ （初回認証の場合は  $n = 1$  のため  $N_2$ ）、認証情報  $A'_n$ （初回認証の場合は  $n = 1$ ）、 $A$ 、 $B'$ 、 $B$ 、認証子  $a$ 、 $b$  である。また、サーバ 3 が保存している情報は、図 8 に示した初期登録時にユーザ 2 から送信された秘密鍵  $K$  および認証情報  $A'_n$ 、 $A_n$ （初回認証の場合は  $n = 1$ ）である。

【0115】

まず、ユーザ 2 は、保存している  $N_{n+1}$  から、下記式 (48) に示す  $b$  を算出し（ステップ S351）、保存している  $b$  と等しいかを判断する（ステップ S352）。このとき、 $b$  の算出に用いる  $ID$ 、 $S$  は、認証処理の都度ユーザ 2 の利用者に入力を求めてもよいし、ユーザ 2 で保存しておいてもよい。保存している  $b$  と等しい場合は（ステップ S352：Yes）、被認証者としての資格を認証し、ステップ S353 に移行する。一方、保存している  $b$  と等しくない場合は（ステップ S352：No）、被認証者としての資格を認証せず、本フローチャートによる処理を終了する。

$$b = X(ID, S, N_{n+1}) \cdots (48)$$

【0116】

また、ユーザ 2 は、さらに新しい乱数  $N_{n+2}$  を生成、保存し（ステップ S353）、下記式 (49) ~ (51) の手順によって  $C'$  を算出する（ステップ S354）。

$$c = X(ID, S, N_{n+2}) \cdots (49)$$

$$C = F(ID, c) \cdots (50)$$

$$C' = F(ID, C) \cdots (51)$$

【0117】

つぎに、ユーザ 2 は、算出した  $b$ 、 $C'$  および保存している  $K$ 、 $A_n$ 、 $A'_n$ 、 $B'$  を用いて、下記式 (52) ~ (54) に示す  $A_{n+1}$ 、 $B_{n+1}$  を算出する（ステップ S355）。そして、サーバ 3 に  $ID$ 、 $A_{n+1}$ 、 $B_{n+1}$  を送信する（ステップ S356）。また、次回認証子  $b$ 、次回認証情報  $B$  として、算出した次々回認証子  $c$ 、次々回認証情報  $C$  を保存して（ステップ S357）、本フローチャートによる処理を終了する。なお、 $A_{n+1}$ 、 $B_{n+1}$  の算出方法は、実施の形態 1 で説明したように、さまざまなバリエーションが存在するが、以下では、その一例として下記式 (52)、(53) に示す演算をおこなうものとする。

$$B' = B' \text{ XOR } A'_n \cdots (52)$$

$$(B' + K) \text{ XOR } A \cdots (53)$$

$$A_{n+1} = E_{\{C'\}}(b) \cdots (54)$$

【0118】

サーバ 3 は、ユーザ 2 から受信した  $A_{n+1}$ 、 $B_{n+1}$  および  $ID$  に対応して保存されている認証情報  $A'_n$ 、秘密鍵  $K$  を用いて、下記式 (55)、(56) の手順によって  $A_n$  を算出する（ステップ S358）。

$$B' = B' \text{ XOR } A'_n \cdots (55)$$

$$A = (B' + K) \text{ XOR } A_{n+1} \cdots (56)$$

【0119】

そして、サーバ 3 は、算出した  $A$  および保存している  $ID$  を一方向性変換した結果  $F(ID, A)$  が  $A'_n$  に等しいかどうかを判断し（ステップ S359）、等しい場合は（ステップ S359：Yes）、ユーザ 2 に対して被認証者の資格を認証する。つづいて、受信した  $B'$  を用いて暗号化情報  $B_n$  を復号し（この処理を  $D_{\{B'\}}(B_n)$  と示す）、下記式 (57) に示す  $a$  を算出する（ステップ S360）。

$$a = D_{\{B'\}}(B_n) \cdots (57)$$

【0120】

そして、算出した  $a$  を  $ID$  とともに一方向性変換した  $F(ID, a)$  が  $A$  に等しいかを判断する（ステップ S361）。 $F(ID, a)$  が  $A$  に等しい場合は（ステップ S361：Yes）、認証情報  $B'$  が改ざんされていないことが証明され、ユーザ 2 との接続を開始する（ステップ S362）。

【0121】

10

20

30

40

50

また、次回（ $n + 1$ 回目）の認証処理時に用いる認証情報（ $A'_{n+1}$ ）として $B'$ を保存する（ステップS363）。また、次回（ $n + 1$ 回目）の認証に用いる認証情報として、ステップS356でユーザ2から送信された $n_{+1}$ を $n$ の代わりに保存して（ステップS364）、本フローチャートによる処理を終了する。

【0122】

一方、ステップS359において、 $F(ID, A)$ が $A'_n$ に等しくない場合は（ステップS359：No）、認証不成立として、ユーザ2にエラーメッセージを送信して（ステップS365）、本フローチャートによる処理を終了する。また、ステップS361において、 $F(ID, a)$ が $A$ に等しくない場合は（ステップS361：No）、認証情報 $B'$ が改ざんされているとして、ユーザ2にエラーメッセージを送信して（ステップS365）、本フローチャートによる処理を終了する。

10

【0123】

以上説明したように、実施の形態3にかかる認証処理システムによれば、認証処理に用いる認証情報は、被認証者（ユーザ2）のみが保存する認証情報の元となるデータ（パスワード $S$ 、乱数 $N_{n+1}$ ）を用いなければ生成することができない。このため、認証者（サーバ3）に保存されている被認証者に関する情報（ $ID, K, A', n$ ）を第三者が盗んだとしても、第三者は認証処理に用いる認証情報を生成することができず、認証者による認証を受けることができない。

【0124】

また、次回認証情報（ $B'$ ）を用いて暗号化した情報（ $n : E_{\{B'\}}(a)$ ）から、今回の認証に用いる今回認証情報の元となるデータ（ $a : D_{\{B'\}}(n)$ ）が算出されるかを確認することによって、第三者による送信情報の偽造を検出することができる。

20

【0125】

さらに、認証に用いるデータ（ $A'$ ）と、送付情報の偽造の検出に用いるデータ（ $A$ ）とを異なるデータにすることによって、認証処理の安全性を高めることができる。

【0126】

以上説明したように、本発明にかかる認証処理方法、認証処理プログラム、記録媒体および認証処理装置では、認証処理に用いるデータは、それぞれマスク処理されて送受信される。このため、認証処理に用いるデータの第三者への漏洩を防ぐことができる。また、今回認証情報の元となるデータ $a$ に一方方向性変換を適用したデータと、今回認証情報 $A$ とが等しいか否かを検証することによって、被認証者の資格を認証することができる。

30

【0127】

また、本発明にかかる認証処理方法、認証処理プログラム、記録媒体および認証処理装置では、認証者に送付された情報の関係検証に加え、今回認証情報の元となるデータ $a$ に一方方向性変換を適用したデータと今回認証情報 $A$ が等しいかを検証する。これにより、送付情報が正当な被認証者によって生成されたか否かを検知することができ、認証者側に登録されている今回認証情報 $A$ が盗まれたとしても、認証者本人以外による不正な認証を防止することができる。

【0128】

なお、本実施の形態で説明した認証処理方法は、あらかじめ用意されたプログラムをパーソナル・コンピュータやワークステーション等のコンピュータで実行することにより実現することができる。このプログラムは、ハードディスク、フレキシブルディスク、CD-ROM、MO、DVD等のコンピュータで読み取り可能な記録媒体に記録され、コンピュータによって記録媒体から読み出されることによって実行される。またこのプログラムは、インターネット等のネットワークを介して配布することが可能な伝送媒体であってもよい。

40

【産業上の利用可能性】

【0129】

以上のように、本発明にかかる認証処理方法、認証処理プログラム、記録媒体および認

50

証処理装置は、被認証者の利用資格、通信資格などの認証に有用であり、特に、情報通信システムや鍵の開閉システムなどに適している。

【図面の簡単な説明】

【0130】

【図1】実施の形態にかかる認証処理システムのシステム構成を示す説明図である。

【図2】認証処理システムを構成するユーザ、サーバのハードウェア構成の一例を示すブロック図である。

【図3】認証処理システムを構成するユーザ、サーバの機能的構成を示すブロック図である。

【図4】ユーザの初期登録処理の手順を示すフローチャートである。

10

【図5】初回（ $n = 1$ ）以降、 $n$ 回目の認証処理の手順を示すフローチャートである。

【図6】ユーザの初期登録処理の手順を示すフローチャートである。

【図7】初回（ $n = 1$ ）以降、 $n$ 回目の認証処理の手順を示すフローチャートである。

【図8】ユーザの初期登録の処理手順を示すフローチャートである。

【図9】初回（ $n = 1$ ）以降、 $n$ 回目の認証処理の手順を示すフローチャートである。

【図10】SAS-2認証方式におけるユーザ認証の処理工程を示すフローチャートである。

【図11】SAS-2認証方式におけるユーザ認証の処理工程を示すフローチャートである。

【符号の説明】

20

【0131】

1 認証処理システム

2 a ~ 2 f ユーザ

3 サーバ

4 ネットワーク

3 1 生成部

3 2 送付部

3 3 算出部

3 4 送信部

4 1 取得部

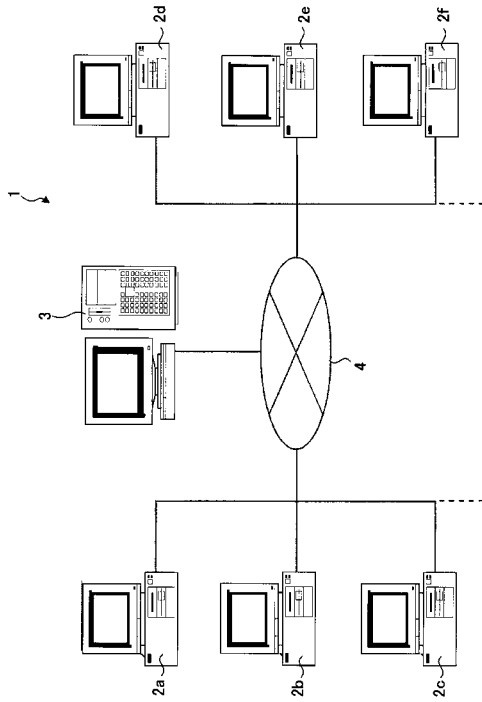
4 2 受信部

4 3 算出部

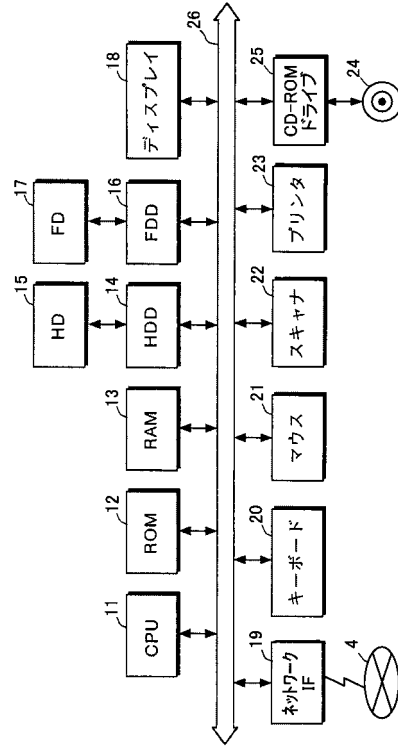
4 4 判断部

30

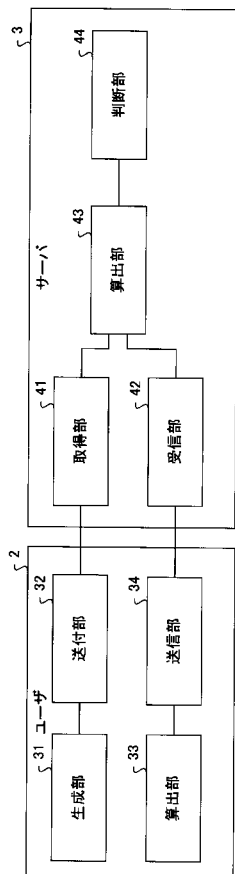
【 図 1 】



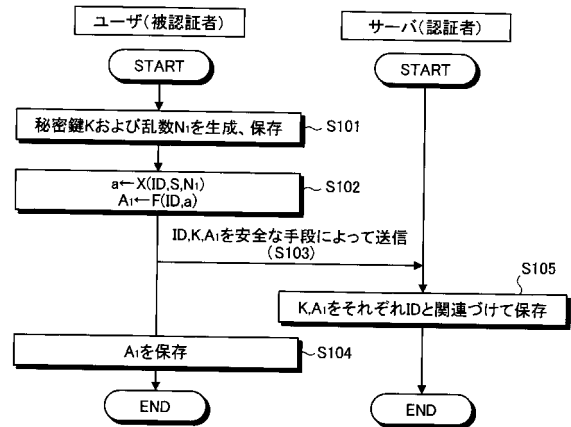
【 図 2 】



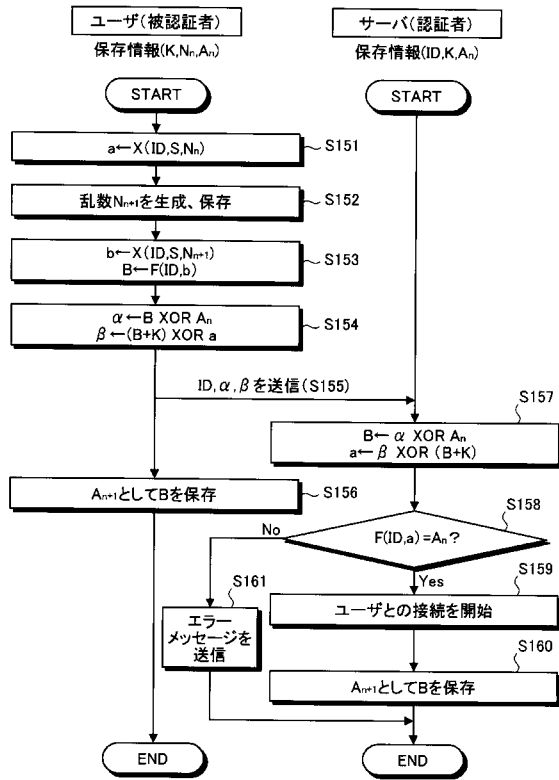
【 図 3 】



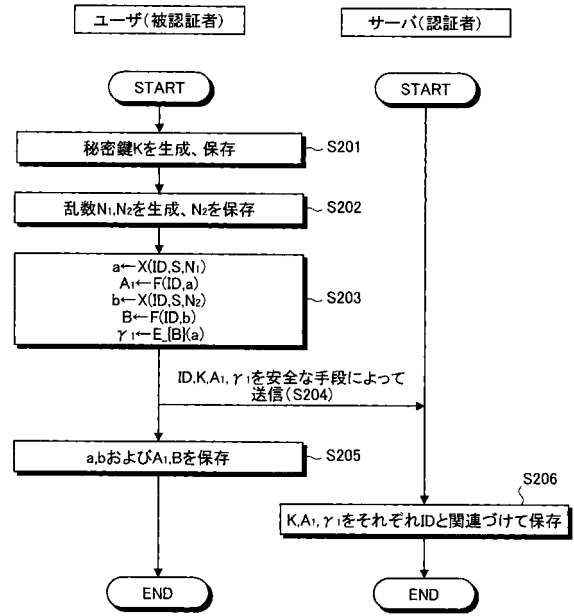
【 図 4 】



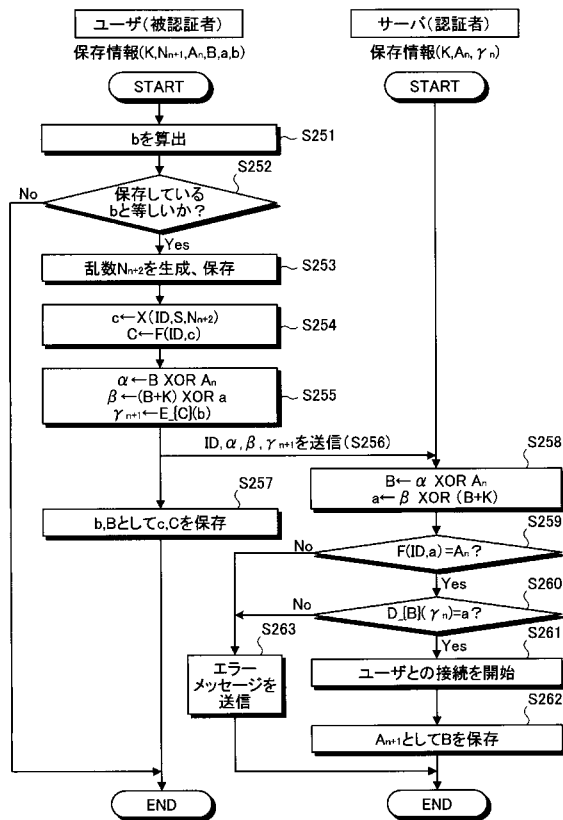
【 図 5 】



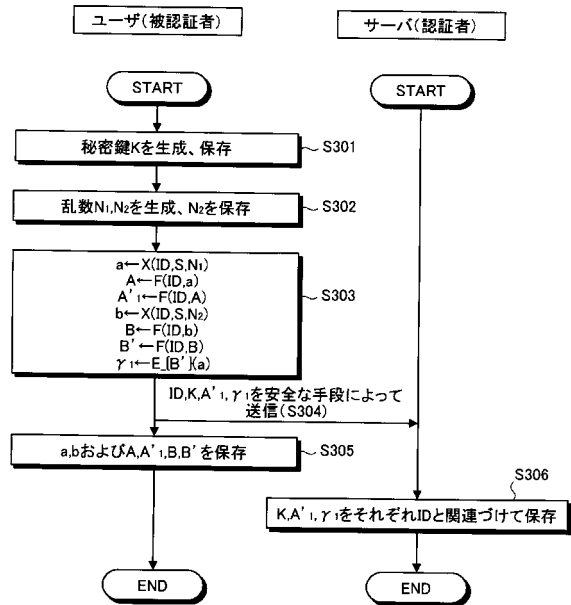
【 図 6 】



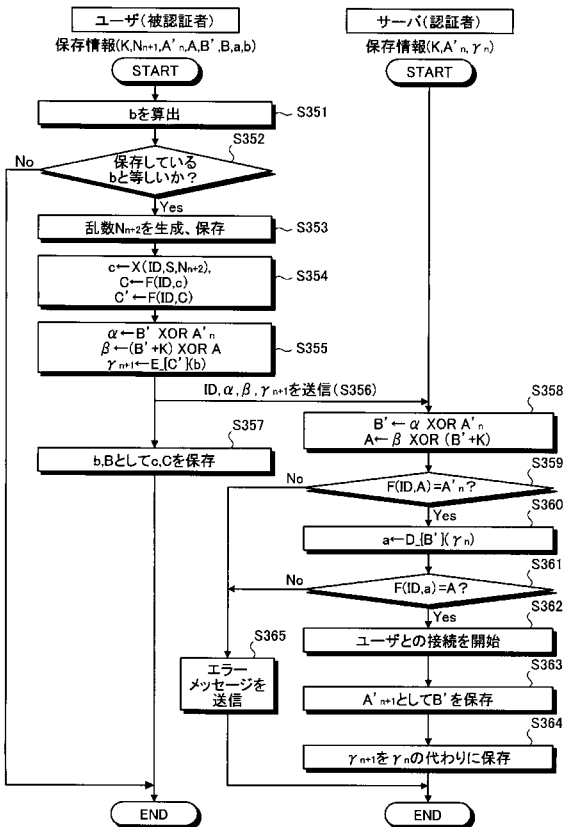
【 図 7 】



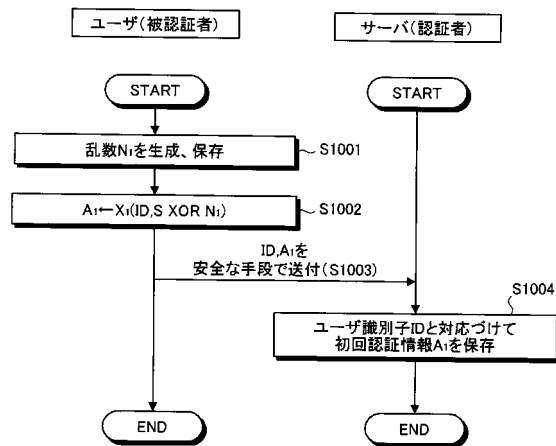
【 図 8 】



【 図 9 】



【 図 10 】



【 図 11 】

