(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7619446号
(P7619446)

(45)発行日　令和7年1月22日(2025.1.22)

(24)登録日　令和7年1月14日(2025.1.14)

(51)国際特許分類
Ｈ０４Ｌ　9/08 (2006.01)

ＦＩ
Ｈ０４Ｌ　9/08　D

請求項の数　5　（全14頁）

(21)出願番号　特願2023-522087(P2023-522087)
(86)(22)出願日　令和3年5月19日(2021.5.19)
(86)国際出願番号　PCT/JP2021/019017
(87)国際公開番号　WO2022/244151
(87)国際公開日　令和4年11月24日(2022.11.24)
審査請求日　令和5年8月29日(2023.8.29)

(73)特許権者　000004226
日本電信電話株式会社
東京都千代田区大手町一丁目５番１号
(74)代理人　110004381
弁理士法人ＩＴＯＨ
(74)代理人　100107766
弁理士　伊東 忠重
(74)代理人　100070150
弁理士　伊東 忠彦
(74)代理人　100124844
弁理士　石原 隆治
(72)発明者　岡野 裕樹
東京都千代田区大手町一丁目５番１号
日本電信電話株式会社内
(72)発明者　小林 鉄太郎

最終頁に続く

(54)【発明の名称】　鍵交換システム、端末、鍵交換方法、及びプログラム

(57)【特許請求の範囲】

【請求項１】

鍵交換を行う複数の端末と、前記端末の認証と前記鍵交換の仲介とを行うサーバとが含まれる鍵交換システムであって、

前記サーバは、

前記端末との間でＯｐｅｎＩＤ　Ｃｏｎｎｅｃｔによる認証連携によって前記認証を行う際に用いられるノンスを生成するノンス生成部と、

トークン制御暗号の公開鍵と秘密鍵とを生成する鍵生成部と、

前記ノンスと、前記公開鍵とを前記端末に送信する第１の送信部と、

前記秘密鍵と、前記ノンスから生成された第１のトークンとを用いて、前記端末から受信した暗号文を復号することにより平文のメッセージを生成する復号部と、

前記メッセージに含まれる認証連携情報に基づいて、前記認証連携が成功したか否かを検証する検証部と、を有し、

前記端末は、

前記公開鍵と、前記サーバから受信したノンスから生成された第２のトークンとを用いて、前記認証連携に用いられる認証連携情報を含むメッセージを暗号化した暗号文を生成する暗号化部と、

前記暗号文を前記サーバに送信する第２の送信部と、

前記ノンスを用いて、前記鍵交換で使用する長期秘密ストリングを生成する長期秘密ストリング生成部と、

を有する鍵交換システム。
【請求項２】
前記長期秘密ストリング生成部は、
前記ノンスを入力とする鍵導出関数又は疑似ランダム関数により前記長期秘密ストリングを生成する、請求項１に記載の鍵交換システム。
【請求項３】
鍵交換を行う他の端末と、各端末の認証と前記鍵交換の仲介とを行うサーバとに通信ネットワークを介して接続される端末であって、
トークン制御暗号の公開鍵であって、かつ、前記サーバで生成された公開鍵と、前記サーバとの間でＯｐｅｎＩＤ　Ｃｏｎｎｅｃｔによる認証連携によって前記認証を行う際に用いられるノンスから生成されたトークンとを用いて、前記認証連携に用いられる認証連携情報を含むメッセージを暗号化した暗号文を生成する暗号化部と、
前記暗号文を前記サーバに送信する送信部と、
前記ノンスを用いて、前記鍵交換で使用する長期秘密ストリングを生成する長期秘密ストリング生成部と、
を有する端末。
【請求項４】
鍵交換を行う複数の端末と、前記端末の認証と前記鍵交換の仲介とを行うサーバとが含まれる鍵交換システムに用いられる鍵交換方法であって、
前記サーバが、
前記端末との間でＯｐｅｎＩＤ　Ｃｏｎｎｅｃｔによる認証連携によって前記認証を行う際に用いられるノンスを生成するノンス生成手順と、
トークン制御暗号の公開鍵と秘密鍵とを生成する鍵生成手順と、
前記ノンスと、前記公開鍵とを前記端末に送信する第１の送信手順と、
前記秘密鍵と、前記ノンスから生成された第１のトークンとを用いて、前記端末から受信した暗号文を復号することにより平文のメッセージを生成する復号手順と、
前記メッセージに含まれる認証連携情報に基づいて、前記認証連携が成功したか否かを検証する検証手順と、を実行し、
前記端末が、
前記公開鍵と、前記サーバから受信したノンスから生成された第２のトークンとを用いて、前記認証連携に用いられる認証連携情報を含むメッセージを暗号化した暗号文を生成する暗号化手順と、
前記暗号文を前記サーバに送信する第２の送信手順と、
前記ノンスを用いて、前記鍵交換で使用する長期秘密ストリングを生成する長期秘密ストリング生成手順と、
を実行する鍵交換方法。
【請求項５】
コンピュータを、請求項１又は２に記載の鍵交換システムに含まれる端末又はサーバとして機能させるためのプログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、鍵交換システム、端末、鍵交換方法、及びプログラムに関する。
【背景技術】
【０００２】
多数の端末間で秘匿通信を行えるようにするために、この秘匿通信の暗号化・復号に利用される共有鍵（セッション鍵）を交換する多者間鍵交換プロトコルと呼ばれる技術が知られている（例えば、非特許文献１及び２等）。非特許文献１及び２に記載されているプロトコルはサーバ支援型の鍵交換プロトコルであり、サーバを介して複数の端末間でセッション鍵を共有する技術である。

【０００３】
非特許文献１及び２に記載されている技術は、前方秘匿性や短期秘密鍵漏洩耐性といった安全性を満たすために、複数の鍵を端末で管理する必要がある。具体的には、公開鍵暗号の秘密鍵又は属性ベース暗号の秘密鍵と長期秘密ストリングとを長期秘密鍵として管理し、短期秘密ストリングを短期秘密鍵として管理する必要がある。特に、長期秘密鍵は端末で半永続的に管理しておくべきものである。ここで、前方秘匿性とは長期秘密鍵が漏洩しても過去の通信内容はなおも安全というものであり、短期秘密鍵漏洩耐性とは鍵交換セッション固有で利用される乱数を生成する乱数生成器が脆弱（つまり、乱数生成器の出力に予測可能性がある場合）であってもそのセッションで共有された鍵はなおも安全というものである。

【先行技術文献】
【非特許文献】
【０００４】
【文献】Kazuki Yoneyama, Reo Yoshida, Yuto Kawahara, Tetsutaro Kobayashi, Hitoshi Fuji, Tomohide Yamamoto, "Multi-Cast Key Distribution: Scalable, Dynamic and Provably Secure Construction," International Conference on Provable Security (ProvSec 2016), LNCS10005, pp.207-226, Nov. 2016.
【文献】Yoneyama, Kazuki, et al. "Exposure-Resilient Identity-Based Dynamic Multi-Cast Key Distribution." IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences 101.6 (2018): 929-944.

【発明の概要】
【発明が解決しようとする課題】
【０００５】
しかしながら、非特許文献１及び２に記載されている技術では、１つのユーザアカウントを用いて複数の不特定の端末で鍵交換を実行する場合には、ユーザアカウントに紐づく長期秘密鍵を予めその端末に格納しておく必要がある。また、不要となった端末からは長期秘密鍵を削除する必要がある。このように、１つのユーザアカウントを用いて複数の不特定の端末で鍵交換を実行する場合には長期秘密鍵の運用・管理コストが大きくなる、という問題がある。この問題に対して、ＯｐｅｎＩＤ Ｃｏｎｎｅｃｔ（以下、「ＯＩＤＣ」ともいう。）と呼ばれる認証連携プロトコルを組み合わせることで、公開鍵暗号の秘密鍵又は属性ベース暗号の秘密鍵を端末が所持する必要のない方式にすることができると考えられる。また、長期秘密ストリングに関しても半永続的に所持する必要のない方式にすることができると考えられる。
【０００６】
なお、長期秘密ストリングを半永続的に所持する必要のない方式にする最も単純な方法としては、鍵交換プロトコルを実行するたびに、長期秘密ストリングを端末で生成することである。一方で、これは、長期秘密ストリングと短期秘密ストリングとを同一の乱数生成器で生成していることになるため、乱数生成器が脆弱であった場合には両方のストリングが漏洩してしまうことになり、短期秘密鍵漏洩耐性を満たすことができない。このため、長期秘密ストリングと短期秘密ストリングはそれぞれ別の乱数生成器から生成する必要がある。
【０００７】
本発明の一実施形態は、上記の点に鑑みてなされたもので、長期秘密鍵の運用・管理コストを削減したサーバ支援型の鍵交換を実現することを目的とする。
【課題を解決するための手段】
【０００８】
上記目的を達成するため、一実施形態に係る鍵交換システムは、鍵交換を行う複数の端末と、前記端末の認証と前記鍵交換の仲介とを行うサーバとが含まれる鍵交換システムであって、前記サーバは、前記端末との間でＯｐｅｎＩＤ Ｃｏｎｎｅｃｔによる認証連携によって前記認証を行う際に用いられるノンスを生成するノンス生成部と、トークン制御

暗号の公開鍵と秘密鍵とを生成する鍵生成部と、前記ノンスと、前記公開鍵とを前記端末に送信する第１の送信部と、前記秘密鍵と、前記端末から受信したトークンとを用いて、前記端末から受信した暗号文を復号する復号部と、を有し、前記端末は、前記公開鍵と、前記ノンスから生成されたトークンとを用いて、所定のデータを暗号化した暗号文を生成する暗号化部と、前記暗号文を前記サーバに送信する第２の送信部と、前記ノンスを用いて、前記鍵交換で使用する長期秘密ストリングを生成する長期秘密ストリング生成部と、を有する。

【発明の効果】

【０００９】

長期秘密鍵の運用・管理コストを削減したサーバ支援型の鍵交換を実現することができる。

【図面の簡単な説明】

【００１０】

【図１】本実施形態に係る鍵交換システムの全体構成の一例を示す図である。

【図２】本実施形態に係る端末の機能構成の一例を示す図である。

【図３】本実施形態に係るサーバの機能構成の一例を示す図である。

【図４】本実施形態に係る認証連携（インプリシットフロー）及び鍵交換処理の一例を示すシーケンス図である。

【図５】本実施形態に係る認証連携（認可コードフロー）及び鍵交換処理の一例を示すシーケンス図である。

【図６】コンピュータのハードウェア構成の一例を示す図である。

【発明を実施するための形態】

【００１１】

以下、本発明の一実施形態について説明する。本実施形態では、ＯＩＤＣを組み合わせることで、長期秘密鍵の運用・管理コストを削減したサーバ支援型の鍵交換を実現することができる鍵交換システム１について説明する。なお、サーバ支援型の鍵交換としては、非特許文献１又は２に記載されている技術を想定する。ＩＤＣについては、例えば、参考文献１「OpenID Connect Core 1.0 incorporating errata set 1, インターネット＜ＵＲＬ：http://openid-foundation-japan.github.io/openid-connect-core-1_0.ja.html＞」等を参照されたい。

【００１２】

＜準備＞

まず、本実施形態で利用する暗号方式や関数を準備する。

【００１３】

≪公開鍵暗号≫

公開鍵暗号は、以下の３つのアルゴリズム（ＫｅｙＧｅｎ，Ｅｎｃ，Ｄｅｃ）で構成される。

【００１４】

ＫｅｙＧｅｎ（1^{κ}）→（ｐｋ，ｓｋ）：セキュリティパラメータκ長の１ビット列1^{κ}を入力とし、公開鍵ｐｋと秘密鍵ｓｋの鍵ペア（ｐｋ，ｓｋ）を出力する鍵生成アルゴリズム。

【００１５】

Ｅｎｃ（ｐｋ，ｍ）→Ｃ：公開鍵ｐｋとメッセージmを入力とし、暗号文Ｃを出力する暗号化アルゴリズム。

【００１６】

Ｄｅｃ（ｓｋ，Ｃ）→m'：秘密鍵ｓｋと暗号文Ｃを入力とし、メッセージm'を出力する復号アルゴリズム。

【００１７】

公開鍵暗号は、正当性として以下の条件も必要とする。

【００１８】

正当性条件：任意のセキュリティパラメータκ、任意の鍵ペア（ｐｋ，ｓｋ）←ＫｅｙＧｅｎ（１$^{\kappa}$）、任意のメッセージｍに対して、Ｄｅｃ（ｓｋ，Ｅｎｃ（ｐｋ，ｍ））＝ｍが成り立つ。

【００１９】

≪トークン制御公開鍵暗号≫

トークン制御公開鍵暗号は、以下の３つのアルゴリズム（ＴＫｅｙＧｅｎ，ＴＥｎｃ，ＴＤｅｃ）で構成される。

【００２０】

ＴＫｅｙＧｅｎ（１$^{\kappa}$）→（ｐｋ，ｓｋ）：セキュリティパラメータκ長の１ビット列１$^{\kappa}$を入力とし、公開鍵ｐｋと秘密鍵ｓｋの鍵ペア（ｐｋ，ｓｋ）を出力する鍵生成アルゴリズム。

【００２１】

ＴＥｎｃ（ｐｋ，ｍ，ｔｏｋｅｎ）→Ｃ：公開鍵ｐｋとメッセージｍとトークンｔｏｋｅｎを入力とし、暗号文Ｃを出力する暗号化アルゴリズム。

【００２２】

ＴＤｅｃ（ｓｋ，Ｃ，ｔｏｋｅｎ）→ｍ'：秘密鍵ｓｋと暗号文Ｃとトークンｔｏｋｅｎを入力とし、メッセージｍ'を出力する復号アルゴリズム。

【００２３】

トークン制御公開鍵暗号は、正当性として以下の条件も必要とする。

【００２４】

正当性条件：任意のセキュリティパラメータκ、任意の鍵ペア（ｐｋ，ｓｋ）←ＴＫｅｙＧｅｎ（１$^{\kappa}$）、任意のトークンｔｏｋｅｎ、任意のメッセージｍに対して、ＴＤｅｃ（ｓｋ，ＴＥｎｃ（ｐｋ，ｍ，ｔｏｋｅｎ），ｔｏｋｅｎ）＝ｍが成り立つ。

【００２５】

なお、トークン制御公開鍵暗号の一例としては、例えば、参考文献２「Galindo, David, and Javier Herranz. "A generic construction for tokencontrolled public key encryption." International Conference on Financial Cryptography and Data Security. Springer, Berlin, Heidelberg, 2006.」に記載されている暗号方式等が挙げられる。

【００２６】

≪鍵導出関数≫

鍵導出関数ＫＤＦ（ｘ，ｓ）は文字列ｘとソルトｓを入力とし、鍵Ｋを出力する関数であり、任意の文字列ｘに対する出力Ｋが、同じ鍵空間から一様ランダムに抽出された鍵Ｋ'と計算量的に識別困難であるような関数のことである。

【００２７】

なお、鍵導出関数の一例としては、例えば、参考文献３「Moriarty, K. M., B. Kaliski, and Andreas Rusch. "RFC 8018: PKCS#5: Password-Based Cryptography Specification Version 2.1." Internet Activities Board (2017).」に記載されているＰＢＫＤＦ２等が挙げられる。

【００２８】

≪疑似ランダム関数≫

疑似ランダム関数ＰＲＦ（ｋ，ｓ）は鍵ｋと文字列ｓを入力とし、鍵Ｋを出力する関数であり、ＰＲＦの出力と、ＰＲＦの右側の入力と同じ定義域を持ち、かつ、ＰＲＦと同じ値域を持つ任意の関数の出力とが計算量的に識別困難であるような関数のことである。

【００２９】

なお、疑似ランダム関数の一例としては、例えば、参考文献４「Krawczyk, Hugo, Mihir Bellare, and Ran Canetti. "RFC2104: HMAC: Keyed-hashing for message authentication." (1997).」に記載されているＨＭＡＣ等が挙げられる。

【００３０】

＜全体構成＞

次に、本実施形態に係る鍵交換システム１の全体構成について、図１を参照しながら説明する。図１は、本実施形態に係る鍵交換システム１の全体構成の一例を示す図である。

【００３１】

図１に示すように、本実施形態に係る鍵交換システム１には、複数の端末１０と、サーバ２０と、ＩＤプロバイダ３０とが含まれる。これらは、例えば、インターネット等の通信ネットワークＮを介して通信可能に接続される。なお、以下では、鍵交換を行う各端末１０の各々を「端末１０－１」、・・・、「端末１０－Ｎ」とする。ここで、Ｎ（ただし、Ｎ≧２）は端末の総数である。

【００３２】

端末１０は、１以上の他の端末１０との間でサーバ支援型の鍵交換プロトコルにより鍵交換を行うユーザ端末である。なお、端末１０としては、例えば、汎用サーバ、ＰＣ（パーソナルコンピュータ）、スマートフォン、タブレット端末、ウェアラブルデバイス、車載器、産業用機器、家電製品、ロボット等といった各種装置や機器等が挙げられる。

【００３３】

サーバ２０は、サーバ支援型の鍵交換プロトコルにより複数の端末１０間で鍵交換を行う際にその鍵交換を支援する（つまり、複数の端末１０間での鍵交換を仲介する）サーバである。ここで、複数の端末１０間で鍵交換を行う際に、サーバ２０は、これらの各端末１０を認証（署名検証、暗号化等）する必要がある。本実施形態では、この認証をＯＩＤＣで利用されるノンスをトークンとしたトークン制御公開鍵暗号により行うと共に、当該ノンスから長期秘密ストリングを生成する。これにより、各端末１０は、公開鍵暗号の秘密鍵又は属性ベース暗号の秘密鍵を保持する必要がなくなると共に、長期秘密ストリングも半永続的に所持する必要はなくなる。このため、長期秘密鍵の運用・管理コストを削減したサーバ支援型の鍵交換が実現される。

【００３４】

なお、上記に加えて、本実施形態では、ＯＰが要求する任意の認証方式によって各端末１０の認証が可能になると共に、ＯＩＤＣのＩＤトークンの有効期間内（又は、サーバ２０が指定した有効期間内）では同一のノンスを使用したトークン制御公開鍵暗号により認証を行うことができるため、当該有効期間内の各鍵交換セッション実行時では再認証を行う必要もなくなる。

【００３５】

ＩＤプロバイダ３０は、ＯＩＤＣのＯＰ（OpenID Provider）として機能するサーバ等である。ＩＤプロバイダ３０は、予め決められた任意の認証方式によるユーザ認証を端末１０に要求する。

【００３６】

＜機能構成＞

次に、本実施形態に係る端末１０及びサーバ２０の機能構成について説明する。

【００３７】

≪端末１０≫

本実施形態に係る端末１０の機能構成について、図２を参照しながら説明する。図２は、本実施形態に係る端末１０の機能構成の一例を示す図である。

【００３８】

図２に示すように、本実施形態に係る端末１０は、認証連携部１０１と、鍵ペア生成部１０２と、暗号化部１０３と、復号部１０４と、長期秘密ストリング生成部１０５とを有する。これら各部は、例えば、端末１０にインストールされた１以上のプログラムが、ＣＰＵ（Central Processing Unit）等のプロセッサに実行させる処理により実現される。

【００３９】

また、本実施形態に係る端末１０は、記憶部１０６を有する。記憶部１０６は、例えば、ＨＤＤ（Hard Disk Drive）やＳＳＤ（Solid State Drive）、フラッシュメモリ等の各種メモリ装置により実現される。

【００４０】

認証連携部１０１は、ＯＩＤＣにより認証連携を行うための各種処理を実行する。鍵ペア生成部１０２は、鍵生成アルゴリズムＫｅｙＧｅｎを実行し、自身の公開鍵と秘密鍵の鍵ペアを生成する。暗号化部１０３は、ＯＩＤＣで利用されるノンスのハッシュ値をトークンとして暗号化アルゴリズムＴＥｎｃを実行し、暗号文を生成する。復号部１０４は、復号アルゴリズムＤｅｃを実行し、暗号文を復号する。長期秘密ストリング生成部１０５は、ＯＩＤＣで利用されるノンスから長期秘密ストリングを生成する。記憶部１０６は、上記の各部が各種処理を実行するために必要な情報やその実行結果等（例えば、各種鍵、ノンス、ＩＤトークン、長期秘密ストリング等）を記憶する。

【００４１】

≪サーバ２０≫

本実施形態に係るサーバ２０の機能構成について、図３を参照しながら説明する。図３は、本実施形態に係るサーバ２０の機能構成の一例を示す図である。

【００４２】

図３に示すように、本実施形態に係るサーバ２０は、認証連携部２０１と、鍵ペア生成部２０２と、暗号化部２０３と、復号部２０４とを有する。これら各部は、例えば、サーバ２０にインストールされた１以上のプログラムが、ＣＰＵ等のプロセッサに実行させる処理により実現される。

【００４３】

また、本実施形態に係るサーバ２０は、記憶部２０５を有する。記憶部２０５は、例えば、ＨＤＤやＳＳＤ、フラッシュメモリ等の各種メモリ装置により実現される。なお、記憶部２０５は、例えば、サーバ２０と通信ネットワークを介して接続される記憶装置等により実現されてもよい。

【００４４】

認証連携部２０１は、ＯＩＤＣにより認証連携を行うための各種処理を実行する。特に、認証連携部２０１は、ＯＩＤＣで利用されるノンスを生成する。鍵ペア生成部２０２は、鍵生成アルゴリズムＴＫｅｙＧｅｎを実行し、サーバの公開鍵と秘密鍵の鍵ペアを生成する。暗号化部２０３は、暗号化アルゴリズムＥｎｃを実行し、暗号文を生成する。復号部２０４は、復号アルゴリズムＴＤｅｃを実行し、暗号文を復号する。このとき、復号部２０４は、自身が生成したノンスのハッシュ値を用いて暗号文を復号する。記憶部２０５は、上記の各部が各種処理を実行するために必要な情報やその実行結果等（例えば、各種鍵、ノンス、ＩＤトークン等）を記憶する。

【００４５】

＜認証連携及び鍵交換処理＞

以下では、本実施形態に係る認証連携及び鍵交換処理について説明する。ここで、非特許文献１及び２に記載されているサーバ支援型の鍵交換プロトコルでは、上述したように、各端末１０が保持する長期秘密鍵として公開鍵暗号又は属性ベース暗号の秘密鍵と長期秘密ストリングｓｔ及びｓｔ'とが存在する。これらの長期秘密ストリングはＤｉｓｔ／Ｊｏｉｎ／Ｌｅａｖｅフェーズにおいて、各フェーズで生成する短期秘密ストリングと合わせてねじれ疑似ランダム関数の入力に使用される。なお、非特許文献１では公開鍵暗号の秘密鍵と長期秘密ストリングｓｔ及びｓｔ'とが長期秘密鍵であり、非特許文献２では属性ベース暗号の秘密鍵と長期秘密ストリングｓｔ及びｓｔ'とが長期秘密鍵である。

【００４６】

上記の長期秘密ストリングｓｔ及びｓｔ'はセットアップ時に各端末１０で生成されるが、本実施形態では、これらの長期秘密ストリングｓｔ及びｓｔ'をノンスから生成する。したがって、各端末１０は長期秘密ストリングｓｔ及びｓｔ'を半永続的に保持するのではなく、ＯＩＤＣのＩＤトークンの有効期間（又は、サーバ２０が指定した有効期間）毎に長期秘密ストリングｓｔ及びｓｔ'を再生成し、その期間の間だけ保持することになる。ただし、１度生成した長期秘密ストリングｓｔ及びｓｔ'を半永続的に保持することとしてもよい。

【００４７】

ＯＩＤＣにはインプリシットフローと呼ばれる認証フローと認可コードフローと呼ばれる認証フローとが存在する。このため、以下では、認証フローがインプリシットフローである場合と認可コードフローである場合のそれぞれについて説明する。認証フローがインプリシットフローである場合は端末１０とサーバ２０がＲＰ（Relying Party）に相当し、認可コードフローである場合はサーバ２０がＲＰに相当する。

【００４８】

≪認証連携がインプリシットフローである場合≫

認証フローがインプリシットフローである場合の認証連携及び鍵交換処理について、図４を参照しながら説明する。図４は、本実施形態に係る認証連携（インプリシットフロー）及び鍵交換処理の一例を示すシーケンス図である。

【００４９】

端末１０の認証連携部１０１は、認証要求をサーバ２０に送信する（ステップＳ１０１）。

【００５０】

サーバ２０の認証連携部２０１は、認証要求を受信すると、ＯＩＤＣで利用するノンスｎｏｎｃｅを生成する（ステップＳ１０２）。また、サーバ２０の鍵ペア生成部２０２は、ＴＫｅｙＧｅｎ（1^{κ}）→（pk_S，sk_S）により公開鍵pk_Sと秘密鍵sk_Sの鍵ペア（pk_S，sk_S）を生成する（ステップＳ１０３）。続いて、サーバ２０の認証連携部２０１は、ＩＤプロバイダ３０へのリダイレクト指示を当該端末１０に送信する（ステップＳ１０４）。このとき、認証連携部２０１は、ノンスｎｏｎｃｅと公開鍵pk_Sとをリダイレクト指示に含める。

【００５１】

端末１０の認証連携部１０１は、リダイレクト指示を受信すると、ＩＤプロバイダ３０にリダイレクトし、このＩＤプロバイダ３０が要求する認証方式によってユーザ認証を行う（ステップＳ１０５）。このとき、認証連携部１０１は、ユーザ認証の際にノンスｎｏｎｃｅをＩＤプロバイダ３０に送信する。なお、ＩＤプロバイダ３０が要求する認証方式としては、例えば、「ＩＤ・パスワード」、「ＳＭＳ認証」、「指紋認証」、「多要素認証」等といった様々な認証方式が挙げられる。

【００５２】

上記のユーザ認証に成功した場合、ＩＤプロバイダ３０から端末１０に対して、ＩＤプロバイダ３０による署名付き、かつ、ノンス付きのＩＤトークンが返信される（ステップＳ１０６）。

【００５３】

端末１０の鍵ペア生成部１０２は、ＩＤプロバイダ３０からＩＤトークンを受信すると、ＫｅｙＧｅｎ（1^{κ}）→（pk_C，sk_C）により公開鍵pk_Cと秘密鍵sk_Cの鍵ペア（pk_C，sk_C）を生成する（ステップＳ１０７）。次に、端末１０の暗号化部１０３は、ノンスｎｏｎｃｅを所定のハッシュ関数（例えば、ＳＨＡ－２５６等）に入力することで得られたハッシュ値をトークンｔｏｋｅｎとして、ＩＤトークンと公開鍵pk_Cを含むメッセージmを公開鍵pk_Sで暗号化した暗号文Ｃを生成する（ステップＳ１０８）。すなわち、暗号化部１０３は、ＴＥｎｃ（pk_S，ｍ，ｔｏｋｅｎ）→Ｃにより暗号文Ｃを生成する。そして、端末１０の暗号化部１０３は、暗号文Ｃをサーバ２０に送信する（ステップＳ１０９）。

【００５４】

サーバ２０の復号部２０４は、暗号文Ｃを受信すると、ノンスｎｏｎｃｅを所定のハッシュ関数（例えば、ＳＨＡ－２５６等）に入力することで得られたハッシュ値をトークンｔｏｋｅｎとして、当該暗号文Ｃを秘密鍵sk_Sで復号する（ステップＳ１１０）。すなわち、復号部２０４は、ＴＤｅｃ（sk_S，Ｃ，ｔｏｋｅｎ）→ｍにより暗号文Ｃを復号してメッセージmを得る。これにより、このメッセージmからＩＤトークンと公開鍵pk_Cが得られる。

【００５５】

サーバ２０の認証連携部２０１は、上記のステップＳ１１０で得られたＩＤトークンの検証を行う（ステップＳ１１１）。すなわち、認証連携部２０１は、当該ＩＤトークンの署名検証を行った後、当該ＩＤトークンに付与されたノンスが上記のステップＳ１０２で生成したノンスｎｏｎｃｅと一致することを検証する。

【００５６】

続いて、上記のステップＳ１１１の検証に成功した場合、端末１０の長期秘密ストリング生成部１０５は、以下の方法１又は方法２のいずれかにより、ノンスｎｏｎｃｅから長期秘密ストリングｓｔ及びｓｔ'を生成する（ステップＳ１１２）。

【００５７】

方法１：ノンスｎｏｎｃｅとソルトを入力とする鍵導出関数ＫＤＦの出力を長期秘密ストリングとする。すなわち、２つのソルトをｓ，ｓ'として、ｓｔ＝ＫＤＦ（ｎｏｎｃｅ，ｓ），ｓｔ'＝ＫＤＦ（ｎｏｎｃｅ，ｓ'）により長期秘密ストリングｓｔ及びｓｔ'を生成する。なお、ソルトｓ，ｓ'としては、例えば、ｓ＝'０００１'，ｓ'＝'０００２'等とすればよい。

【００５８】

方法２：ノンスｎｏｎｃｅと文字列を入力とする疑似ランダム関数ＰＲＦの出力を長期秘密ストリングとする。すなわち、２つの文字列をｓ，ｓ'として、ｓｔ＝ＰＲＦ（ｎｏｎｃｅ，ｓ），ｓｔ'＝ＰＲＦ（ｎｏｎｃｅ，ｓ'）により長期秘密ストリングｓｔ及びｓｔ'を生成する。なお、文字列ｓ，ｓ'としては、例えば、ｓ＝'０００１'，ｓ'＝'０００２'等とすればよい。

【００５９】

このように、端末１０で使われる乱数生成器とは異なる乱数生成器から生成した乱数（つまり、ノンスｎｏｎｃｅ）から長期秘密ストリングを生成することで、仮に端末１０の乱数生成器が脆弱であっても長期秘密ストリングが漏洩することがないため、安全な鍵交換が実現できる。また、ＯＩＤＣのノンスｎｏｎｃｅを用いるため、長期秘密ストリングを生成するための余分な通信や計算を行う必要がなく、長期秘密ストリングを効率的に生成することが可能である。

【００６０】

そして、上記のステップＳ１１２で長期秘密ストリングｓｔ及びｓｔ'が生成された場合、端末１０とサーバ２０は鍵交換を行う（ステップＳ１１３）。この鍵交換の詳細については後述する。

【００６１】

≪認証連携が認可コードフローである場合≫

認証連携が認可コードフローである場合の認証連携及び鍵交換処理について、図５を参照しながら説明する。図５は、本実施形態に係る認証連携（認可コードフロー）及び鍵交換処理の一例を示すシーケンス図である。

【００６２】

端末１０の認証連携部１０１は、認証要求をサーバ２０に送信する（ステップＳ２０１）。

【００６３】

サーバ２０の認証連携部２０１は、認証要求を受信すると、ＯＩＤＣで利用するノンスｎｏｎｃｅを生成する（ステップＳ２０２）。また、サーバ２０の鍵ペア生成部２０２は、ＴＫｅｙＧｅｎ（1^{κ}）→（pk_S，sk_S）により公開鍵pk_Sと秘密鍵sk_Sの鍵ペア（pk_S，sk_S）を生成する（ステップＳ２０３）。続いて、サーバ２０の認証連携部２０１は、ＩＤプロバイダ３０へのリダイレクト指示を当該端末１０に送信する（ステップＳ２０４）。このとき、認証連携部２０１は、ノンスｎｏｎｃｅと公開鍵pk_Sとをリダイレクト指示に含める。

【００６４】

端末１０の認証連携部１０１は、リダイレクト指示を受信すると、ＩＤプロバイダ３０にリダイレクトし、このＩＤプロバイダ３０が要求する認証方式によってユーザ認証を行

う（ステップＳ２０５）。このとき、認証連携部１０１は、ユーザ認証の際にノンスｎｏｎｃｅをＩＤプロバイダ３０に送信する。

【００６５】

上記のユーザ認証に成功した場合、ＩＤプロバイダ３０から端末１０に対して、認可コードが返信される（ステップＳ２０６）。

【００６６】

端末１０の鍵ペア生成部１０２は、ＩＤプロバイダ３０から認可コードを受信すると、ＫｅｙＧｅｎ（1^{κ}）→（pk_C，sk_C）により公開鍵pk_Cと秘密鍵sk_Cの鍵ペア（pk_C，sk_C）を生成する（ステップＳ２０７）。次に、端末１０の暗号化部１０３は、ノンスｎｏｎｃｅを所定のハッシュ関数（例えば、ＳＨＡ－２５６等）に入力することで得られたハッシュ値をトークンｔｏｋｅｎとして、認可コードと公開鍵pk_Cを含むメッセージｍを公開鍵pk_Sで暗号化した暗号文Ｃを生成する（ステップＳ２０８）。すなわち、暗号化部１０３は、ＴＥｎｃ（pk_S，ｍ，ｔｏｋｅｎ）→Ｃにより暗号文Ｃを生成する。そして、端末１０の暗号化部１０３は、暗号文Ｃをサーバ２０に送信する（ステップＳ２０９）。

【００６７】

サーバ２０の復号部２０４は、暗号文Ｃを受信すると、ノンスｎｏｎｃｅを所定のハッシュ関数（例えば、ＳＨＡ－２５６等）に入力することで得られたハッシュ値をトークンｔｏｋｅｎとして、当該暗号文Ｃを秘密鍵sk_Sで復号する（ステップＳ２１０）。すなわち、復号部２０４は、ＴＤｅｃ（sk_S，Ｃ，ｔｏｋｅｎ）→ｍにより暗号文Ｃを復号してメッセージｍを得る。これにより、このメッセージｍから認可コードと公開鍵pk_Cが得られる。

【００６８】

次に、サーバ２０の認証連携部２０１は、上記のステップＳ２１０で得られた認可コードをＩＤプロバイダ３０に送信する（ステップＳ２１１）。これにより、ＩＤプロバイダ３０からサーバ２０に対して、ＩＤプロバイダ３０による署名付き、かつ、ノンス付きのＩＤトークンが返信される（ステップＳ２１２）。

【００６９】

サーバ２０の認証連携部２０１は、上記のステップＳ２１２で得られたＩＤトークンの検証を行う（ステップＳ２１３）。すなわち、認証連携部２０１は、当該ＩＤトークンの署名検証を行った後、当該ＩＤトークンに付与されたノンスが上記のステップＳ２０２で生成したノンスｎｏｎｃｅと一致することを検証する。

【００７０】

続いて、上記のステップＳ２１３の検証に成功した場合、端末１０の長期秘密ストリング生成部１０５は、図４のステップＳ１１２と同様に、方法１又は方法２のいずれかにより、ノンスｎｏｎｃｅから長期秘密ストリングｓｔ及びｓｔ'を生成する（ステップＳ２１４）。

【００７１】

そして、上記のステップＳ２１４で長期秘密ストリングｓｔ及びｓｔ'が生成された場合、端末１０とサーバ２０は鍵交換を行う（ステップＳ２１５）。この鍵交換の詳細については後述する。

【００７２】

≪鍵交換処理≫

上記のステップＳ１１３又はステップＳ２１５の鍵交換について説明する。ここでは、上記の非特許文献１及び２に記載されている鍵交換を行う場合について説明する。なお、以下で特に説明を行った処理以外に関しては非特許文献１及び２を参照されたい。

【００７３】

これらの非特許文献１及び２に記載されている鍵交換では、最初にサーバ２０から端末１０にＭＡＣ鍵と属性ベース暗号の鍵が送信される。そこで、このとき、サーバ２０の暗号化部２０３は、これらの鍵を含むメッセージｍ'を公開鍵pk_Cで暗号化した暗号文Ｃを

生成、つまり、Ｅｎｃ（ｐｋ$_{C}$，ｍ'）→Ｃにより暗号文Ｃを生成し、その暗号文Ｃを当該端末１０に送信する。そして、端末１０の復号部１０４は、この暗号文Ｃを復号、つまり、Ｄｅｃ（ｓｋ$_{C}$，Ｃ）→ｍ'によりメッセージｍ'を生成し、このメッセージｍ'からＭＡＣ鍵と属性ベース暗号の鍵を取り出す。その後の処理は、非特許文献１及び２に記載されている処理と同様である。

【００７４】

＜ハードウェア構成＞

最後に、本実施形態に係る端末１０及びサーバ２０のハードウェア構成について説明する。本実施形態に係る端末１０及びサーバ２０は、例えば、図６に示すコンピュータ５００のハードウェア構成により実現される。図６は、コンピュータ５００のハードウェア構成の一例を示す図である。

【００７５】

図６に示すコンピュータ５００は、入力装置５０１と、表示装置５０２と、外部Ｉ／Ｆ５０３と、通信Ｉ／Ｆ５０４と、プロセッサ５０５と、メモリ装置５０６とを有する。これらの各ハードウェアは、それぞれがバス５０７により通信可能に接続される。

【００７６】

入力装置５０１は、例えば、キーボードやマウス、タッチパネル等である。表示装置５０２は、例えば、ディスプレイ等である。なお、コンピュータ５００は、例えば、入力装置５０１及び表示装置５０２のうちの少なくとも一方を有していなくてもよい。

【００７７】

外部Ｉ／Ｆ５０３は、記録媒体５０３ａ等の外部装置とのインタフェースである。コンピュータ５００は、外部Ｉ／Ｆ５０３を介して、記録媒体５０３ａの読み取りや書き込み等を行うことができる。なお、記録媒体５０３ａとしては、例えば、ＣＤ（Compact Disc）、ＤＶＤ（Digital Versatile Disk）、ＳＤメモリカード（Secure Digital memory card）、ＵＳＢ（Universal Serial Bus）メモリカード等が挙げられる。

【００７８】

通信Ｉ／Ｆ５０４は、コンピュータ５００を通信ネットワークに接続するためのインタフェースである。プロセッサ５０５は、例えば、ＣＰＵ等の各種演算装置である。メモリ装置５０６は、例えば、ＨＤＤやＳＳＤ、フラッシュメモリ、ＲＡＭ（Random Access Memory）、ＲＯＭ（Read Only Memory）等の各種記憶装置である。

【００７９】

本実施形態に係る端末１０及びサーバ２０は、図６に示すハードウェア構成を有することにより、上述した認証連携及び鍵交換処理を実現することができる。なお、図６に示すハードウェア構成は一例であって、コンピュータ５００は、例えば、複数のプロセッサを有していたり、複数のメモリ装置を有していたりしてもよく、様々なハードウェア構成を有していてもよい。

【００８０】

本発明は、具体的に開示された上記の実施形態に限定されるものではなく、請求の範囲の記載から逸脱することなく、種々の変形や変更、既知の技術との組み合わせ等が可能である。

【符号の説明】

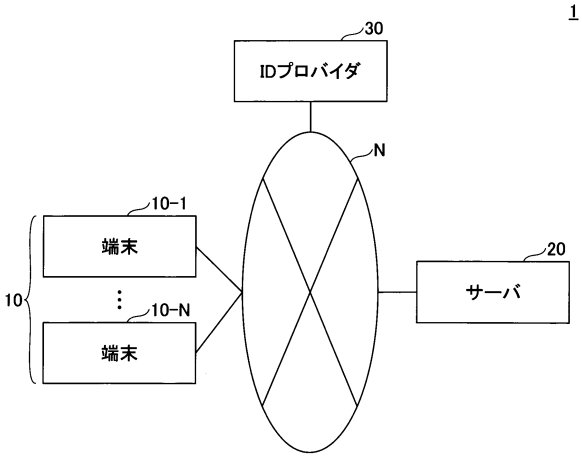
【００８１】

１　　鍵交換システム
１０　端末
２０　サーバ
３０　ＩＤプロバイダ
１０１　認証連携部
１０２　鍵ペア生成部
１０３　暗号化部
１０４　復号部

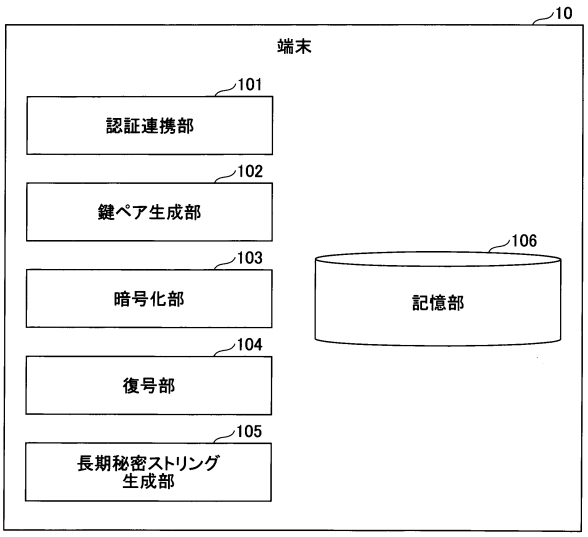
１０５　長期秘密ストリング生成部
１０６　記憶部
２０１　認証連携部
２０２　鍵ペア生成部
２０３　暗号化部
２０４　復号部
２０５　記憶部
Ｎ　通信ネットワーク

【図面】

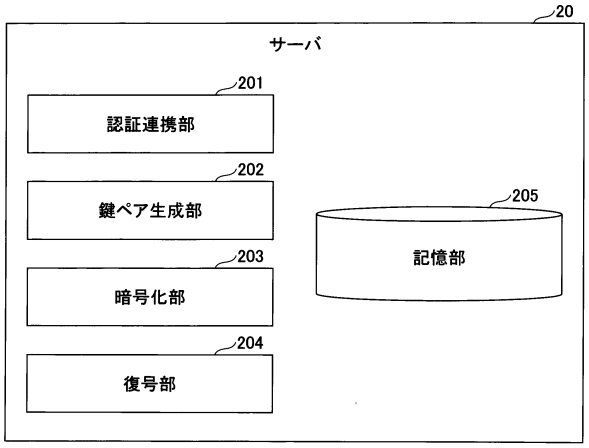
【図１】



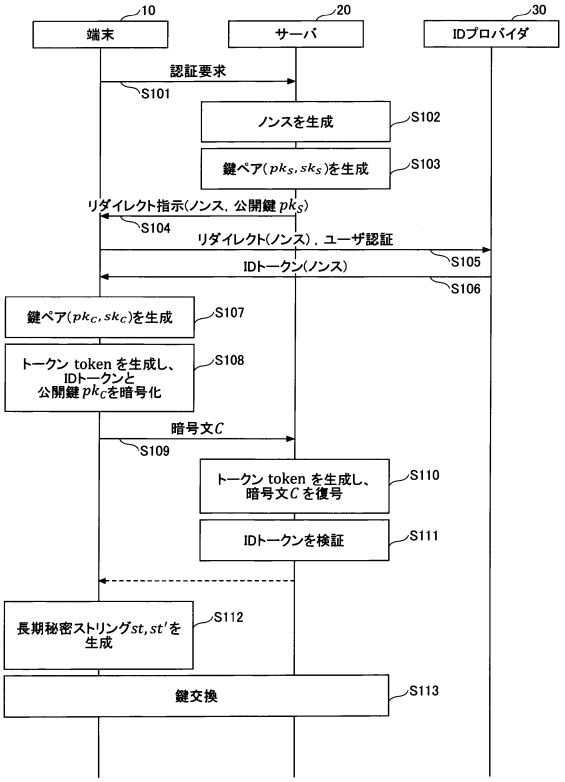
【図２】



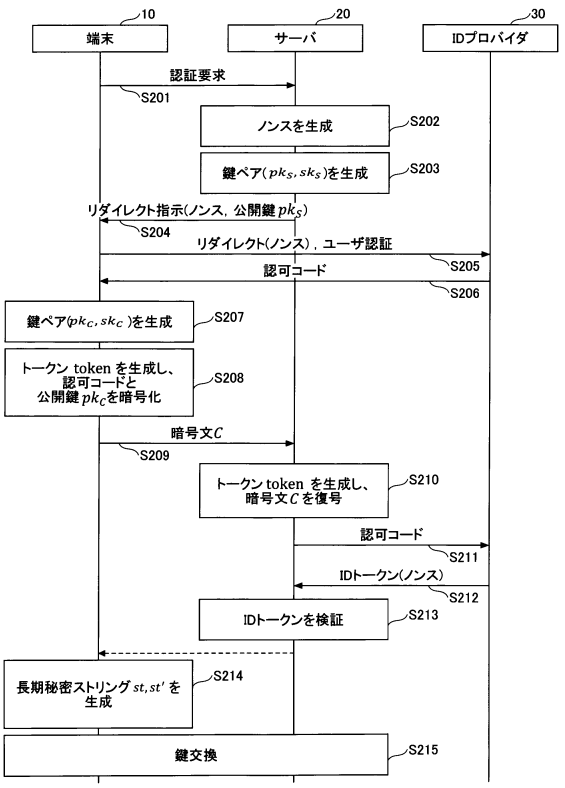
【図３】


20
サーバ
201
認証連携部
202
鍵ペア生成部
203
暗号化部
204
復号部
205
記憶部

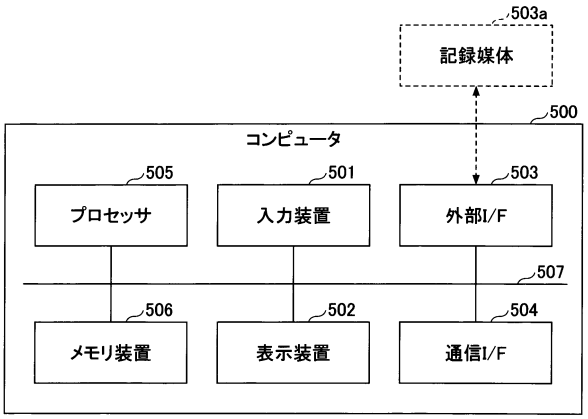
【図４】


10
端末
20
サーバ
30
IDプロバイダ
認証要求
S101
ノンスを生成
S102
鍵ペア(pk_S, sk_S)を生成
S103
リダイレクト指示(ノンス, 公開鍵 pk_S)
S104
リダイレクト(ノンス), ユーザ認証
S105
IDトークン(ノンス)
S106
鍵ペア(pk_C, sk_C)を生成
S107
トークン token を生成し、IDトークンと公開鍵 pk_Cを暗号化
S108
暗号文C
S109
トークン token を生成し、暗号文Cを復号
S110
IDトークンを検証
S111
長期秘密ストリングst, st'を生成
S112
鍵交換
S113

【図５】


10
端末
20
サーバ
30
IDプロバイダ
認証要求
S201
ノンスを生成
S202
鍵ペア(pk_S, sk_S)を生成
S203
リダイレクト指示(ノンス, 公開鍵 pk_S)
S204
リダイレクト(ノンス), ユーザ認証
S205
認可コード
S206
鍵ペア(pk_C, sk_C)を生成
S207
トークン token を生成し、認可コードと公開鍵 pk_Cを暗号化
S208
暗号文C
S209
トークン token を生成し、暗号文Cを復号
S210
認可コード
S211
IDトークン(ノンス)
S212
IDトークンを検証
S213
長期秘密ストリングst, st'を生成
S214
鍵交換
S215

【図６】


503a
記録媒体
500
コンピュータ
505
プロセッサ
501
入力装置
503
外部I/F
507
506
メモリ装置
502
表示装置
504
通信I/F

フロントページの続き

東京都千代田区大手町一丁目５番１号 日本電信電話株式会社内
(72)発明者 村上 啓造
東京都千代田区大手町一丁目５番１号 日本電信電話株式会社内
(72)発明者 奥田 哲矢
東京都千代田区大手町一丁目５番１号 日本電信電話株式会社内

審査官 松平 英
(56)参考文献 特表２０２０－５２００１７（ＪＰ，Ａ）
特開２０１９－１３９５２０（ＪＰ，Ａ）
特開２００８－１３１６５２（ＪＰ，Ａ）
国際公開第２０１９／１９８５１６（WO，Ａ１）
(58)調査した分野 (Int.Cl.，ＤＢ名)
Ｇ０６Ｆ１２／１４
２１／００－２１／８８
Ｇ０９Ｃ １／００－５／００
Ｈ０４Ｋ １／００－３／００
Ｈ０４Ｌ ９／００－９／４０