【公報種別】特許法第１７条の２の規定による補正の掲載
【部門区分】第７部門第３区分
【発行日】平成30年2月1日(2018.2.1)

【公開番号】特開2015-146567(P2015-146567A)
【公開日】平成27年8月13日(2015.8.13)
【年通号数】公開・登録公報2015-051
【出願番号】特願2014-253636(P2014-253636)
【国際特許分類】
Ｈ０４Ｌ　9/32　(2006.01)
Ｈ０４Ｌ　9/08　(2006.01)
Ｇ０６Ｆ　21/44　(2013.01)
【ＦＩ】
Ｈ０４Ｌ　9/00　６７５Ａ
Ｈ０４Ｌ　9/00　６０１Ｂ
Ｇ０６Ｆ　21/44　３５０

【手続補正書】
【提出日】平成29年12月14日(2017.12.14)
【手続補正１】
【補正対象書類名】明細書
【補正対象項目名】００１１
【補正方法】変更
【補正の内容】
【００１１】
本発明は、サーバーとクライアント間のデータグラム転送に関する双方向の相互認証のためのコンピュータにより実施される、システムプロセッサを含むシステムであって、前記システムプロセッサと協働し、第１の乱数を発生する第１の乱数発生器；前記システムプロセッサと協働し、第２の乱数を発生する第２の乱数発生器；前記システムプロセッサと協働し、プロビジョニング処理の間の双方向認証の前に、システムプロセッサの生成コマンド及び送信コマンドを受けて秘密鍵を生成してサーバー及びクライアントに送信する秘密鍵生成器；前記システムプロセッサの転送コマンドを受けて、クライアントの固有ＩＤを含む第１のメッセージをクライアントからサーバーに送信するセッションイニシエータ；前記システムプロセッサと協働して、システムプロセッサの受信コマンドを受けて前記第１のメッセージを受信し、受信したクライアントＩＤと予め格納されたクライアントを識別するためのクライアントＩＤを照合するマッチングエンジンを搭載する受信機；固有時間制限付きセッション鍵を生成し、前記システムプロセッサの送信コマンドを受けて生成したセッション鍵を転送するセッション鍵生成器であって、前記システムプロセッサからのコマンドに応じて動作するセッション鍵タイマーを有し、該セッション鍵タイマー値の満了により、生成したセッション鍵を無効にし、新しいセッションの確立の要件を示すセッション鍵生成器；前記システムプロセッサと協働して、前記セッション鍵を受け取り、第１の乱数発生器によって発生された第１の乱数と前記セッション鍵生成器によって生成された前記セッション鍵を含むチャレンジコードを、システムプロセッサの発生コマンドを受けて生成し、プロセッサの送信コマンドを受けて送信するチャレンジコード生成器；前記チャレンジコード生成器と協働して、システムプロセッサからのコマンドに応じて、前記秘密鍵生成器により秘密鍵とともに生成されたチャレンジコードを受信し、該生成されたチャレンジコードを受信して暗号化し、さらにシステムプロセッサからの送信コマンドを受けて、前記暗号化されたチャレンジコードを特定クライアントに送信する第１の暗号器；前記システムプロセッサと協働して、前記暗号化されたチャレンジコードを受信し、さらに、システムプロセッサからのコマンドに応じて、秘密鍵生成器によって生成

された秘密鍵で暗号化されたチャレンジコードを復号化し、復号化された第１の乱数及びセッション鍵を得る第１の復号器；前記第１の復号器から受信したセッション鍵を格納するリポジトリ；前記システムプロセッサと協働し、前記復号化された第１の乱数とセッション鍵を受信し、さらにシステムプロセッサからの送信コマンドを受けて、復号化された第１の乱数と、第２の乱数発生器によって生成され、セッション鍵で暗号化された第２の乱数を含む第２のメッセージを送信する第２暗号器；前記システムプロセッサと協働し、前記第２のメッセージを受信し、さらにシステムプロセサからのコマンドに応じて、前記第１の乱数と第２の乱数を、セッション鍵生成器によって生成されたセッション鍵を用いて復号化する第２の復号器；前記システムプロセッサからのコマンドに応じて、前記第２のメッセージから復号化された第１の乱数と前記第１の乱数発生器で生成された第１の乱数を比較してクライアントを認証する第１のコンパレータと認証器；クライアントを認証した後、前記システムプロセッサからのコマンドに応じて、受信したセッション鍵生成器によって生成されたセッション鍵が含まれる第２のメッセージの中の第２の乱数を暗号化し、システムプロセッサの送信コマンドを受けて、暗号化された第２の乱数を送信する第３の暗号器；前記システムプロセッサからの受信コマンドに応じて受信し、さらにシステムプロセッサからのコマンドに応じて、リポジトリから受信したセッション鍵で暗号化された第２の乱数を復号化する第３の復号器；及び前記システムプロセッサからのコマンドに応じて、前記復号化した第２の乱数と前記第２の乱数発生器によって発生された第２の乱数を比較し、サーバーの認証と相互認証を達成する第２のコンパレータと認証器；を含み、前記第１の乱数発生器で生成された第１の乱数は、第１のタイマー値を付加した第１の疑似乱数であること、を特徴とする。

【手続補正２】
【補正対象書類名】明細書
【補正対象項目名】００１２
【補正方法】変更
【補正の内容】
【００１２】

本発明の方法は、サーバーとクライアント間のデータグラム転送に関する双方向の相互認証のためのコンピュータにより実施される、システム処理コマンドを含む方法であって、第１の乱数発生器を利用して、第１の乱数を発生するステップ；第２の乱数発生器を利用して、第２の乱数を発生するステップ；秘密鍵生成器を利用して、システム処理コマンドに応じて秘密鍵を生成するステップ；システム処理コマンドに応じて、双方向認証の前であって、プロビジョニング処理の間に、前記生成された秘密鍵をサーバーとクライアントに送信するステップ；システム処理コマンドに応じて、クライアントの固有ＩＤを含む第１のメッセージを送信するステップ；システム処理コマンドに応じて、前記第１のメッセージを受信し、受信したクライアントＩＤを予め格納されているクライアントＩＤと照合するステップ；前記受信したクライアントＩＤに基づいてクライアントを識別するステップ；セッション鍵生成器を利用して、セッション鍵タイマー値の満了に基づいてセッション鍵を無効にし、満了の上で新しいセッションの確立の要求を示す固有時間制限付きセッション鍵を生成するステップ；セッション鍵を受信し、システム処理コマンドを受けて、第１の乱数発生器によって生成された第１の乱数と、前記セッション鍵生成器により生成された前記セッション鍵を含むチャレンジコードを生成するステップ；前記チャレンジコード生成器により生成されたチャレンジコードを受信し、システム処理コマンドに応じて、前記第１の暗号化部を利用して前記秘密鍵生成器により生成された秘密鍵で前記受信したチャレンジコードを暗号化し、暗号化したチャレンジコードをシステム処理コマンドに応じて送信するステップと；前記暗号化されたチャレンジコードを受信し、システム処理コマンドに応じて、前記第１の復号化器を利用して、前記秘密鍵生成器により生成された秘密鍵で暗号化されたチャレンジコードを復号化し、第１の乱数とセッション鍵を取得するステップ；前記第１の復号器からセッション鍵を受信し、リポジトリに格納するステップ；システムの処理コマンドに応じて、前記復号化された第１の乱数とセッション鍵を

受信し、該復号化された第１の乱数及び前記第２の乱数発生器により発生された第２の乱数を含むセッション鍵で暗号化された第２のメッセージを送信するステップ；システム処理コマンドに応じて、第２のメッセージを受信し、第１の乱数と第２の乱数を、前記セッション鍵生成器により生成されたセッション鍵を用いて復号化するステップ；システムの処理コマンドに応じて、第２のメッセージからの復号化された第１の乱数と第１の乱数発生器によって発生された第１の乱数を比較するステップ；システム処理コマンドを受けて、前記復号化された第１の乱数が前記第１の乱数発生器によって発生された第１の乱数と一致したとき、前記クライアントを認証するステップ；クライアントを認証した後、システムの処理コマンドに応じて、前記セッション鍵発生器によって生成されたセッション鍵とともに受信した第２のメッセージ中の第２の乱数を暗号化し、暗号化された第２の乱数を送信するステップ；システム処理コマンドに応じて受信し、リポジトリから受信したセッション鍵で暗号化された第２の乱数を復号化するステップ；システム処理コマンドに応じて、前記復号化された第２の乱数と前記第２の乱数発生器によって発生された第２の乱数を比較するステップ；及びシステム処理コマンドに応じて、前記復号化された第２の乱数と前記第２の乱数発生器によって発生された第２の乱数が一致するとき、相互認証を確立するためにサーバーを認証するステップ；を含み、前記第１の乱数発生器によって発生された第１の乱数は、第１のタイマー値を付加した第１の疑似乱数であること、を特徴とする。

【手続補正３】
【補正対象書類名】特許請求の範囲
【補正対象項目名】全文
【補正方法】変更
【補正の内容】
【特許請求の範囲】
【請求項１】
サーバーとクライアント間のデータグラム転送に関する双方向の相互認証のためのコンピュータにより実施される、システムプロセッサを含むシステムであって、
前記システムプロセッサと協働し、第１の乱数を発生する第１の乱数発生器；
前記システムプロセッサと協働し、第２の乱数を発生する第２の乱数発生器；
前記システムプロセッサと協働し、プロビジョニング処理の間の双方向認証の前に、システムプロセッサの生成コマンド及び送信コマンドを受けて秘密鍵を生成してサーバー及びクライアントに送信する秘密鍵生成器；
前記システムプロセッサの転送コマンドを受けて、クライアントの固有ＩＤを含む第１のメッセージをクライアントからサーバーに送信するセッションイニシエータ；
前記システムプロセッサと協働して、システムプロセッサの受信コマンドを受けて前記第１のメッセージを受信し、受信したクライアントＩＤと予め格納されたクライアントを識別するためのクライアントＩＤを照合するマッチングエンジンを搭載する受信機；
固有時間制限付きセッション鍵を生成し、前記システムプロセッサの送信コマンドを受けて生成したセッション鍵を転送するセッション鍵生成器であって、前記システムプロセッサからのコマンドに応じて動作するセッション鍵タイマーを有し、該セッション鍵タイマー値の満了により、生成したセッション鍵を無効にし、新しいセッションの確立の要件を示すセッション鍵生成器；
前記システムプロセッサと協働して、前記セッション鍵を受け取り、第１の乱数発生器によって発生された第１の乱数と前記セッション鍵生成器によって生成された前記セッション鍵を含むチャレンジコードを、システムプロセッサの発生コマンドを受けて生成し、プロセッサの送信コマンドを受けて送信するチャレンジコード生成器；
前記チャレンジコード生成器と協働して、システムプロセッサからのコマンドに応じて、前記秘密鍵生成器により秘密鍵とともに生成されたチャレンジコードを受信し、該生成されたチャレンジコードを受信して暗号化し、さらにシステムプロセッサからの送信コマンドを受けて、前記暗号化されたチャレンジコードを特定クライアントに送信する第１の

暗号器；
前記システムプロセッサと協働して、前記暗号化されたチャレンジコードを受信し、さらに、システムプロセッサからのコマンドに応じて、秘密鍵生成器によって生成された秘密鍵で暗号化されたチャレンジコードを復号化し、復号化された第１の乱数及びセッション鍵を得る第１の復号器；
前記第１の復号器から受信したセッション鍵を格納するリポジトリ；
前記システムプロセッサと協働し、前記復号化された第１の乱数とセッション鍵を受信し、さらにシステムプロセッサからの送信コマンドを受けて、復号化された第１の乱数と、第２の乱数発生器によって生成され、セッション鍵で暗号化された第２の乱数を含む第２のメッセージを送信する第２暗号器；
前記システムプロセッサと協働し、前記第２のメッセージを受信し、さらにシステムプロセッサからのコマンドに応じて、前記第１の乱数と第２の乱数を、セッション鍵生成器によって生成されたセッション鍵を用いて復号化する第２の復号器；
前記システムプロセッサからのコマンドに応じて、前記第２のメッセージから復号化された第１の乱数と前記第１の乱数発生器で生成された第１の乱数を比較してクライアントを認証する第１のコンパレータと認証器；
クライアントを認証した後、前記システムプロセッサからのコマンドに応じて、受信したセッション鍵生成器によって生成されたセッション鍵が含まれる第２のメッセージの中の第２の乱数を暗号化し、システムプロセッサの送信コマンドを受けて、暗号化された第２の乱数を送信する第３の暗号器；
前記システムプロセッサからの受信コマンドに応じて受信し、さらにシステムプロセッサからのコマンドに応じて、リポジトリから受信したセッション鍵で暗号化された第２の乱数を復号化する第３の復号器；及び
前記システムプロセッサからのコマンドに応じて、前記復号化した第２の乱数と前記第２の乱数発生器によって発生された第２の乱数を比較し、サーバーの認証と相互認証を達成する第２のコンパレータと認証器；を含み、
前記第１の乱数発生器で生成された第１の乱数は、第１のタイマー値を付加した第１の疑似乱数であること、
を特徴とするシステム。
【請求項２】
請求項１記載のシステムにおいて、前記第１の乱数発生器は、前記システムプロセッサからのコマンドに応じて前記第１のタイマー値を生成する第１のタイマーを有するシステム。
【請求項３】
請求項１記載のシステムにおいて、前記第２の乱数発生器は、前記システムプロセッサからのコマンドに応じて第２のタイマー値を生成する第２のタイマーを有するシステム。
【請求項４】
請求項１記載のシステムにおいて、前記第２の乱数発生器によって発生された第２の乱数は、前記第２のタイマー値を付加した第２の擬似乱数であるシステム。
【請求項５】
請求項１記載のシステムにおいて、前記秘密鍵生成器によって生成された秘密鍵は、前記セッションの開始時に生成された固有の鍵であって、進行中のセッションの間のみ有効であるシステム。
【請求項６】
請求項１記載のシステムにおいて、前記第１の乱数発生器及び第２の乱数発生器によって発生された乱数は、再現不能であり、異なるセッションによって変化するシステム。
【請求項７】
請求項１記載のシステムにおいて、前記クライアントは、前記サーバーがクライアントの実行要求のステータスに応答しないように前記サーバーと通信するシステム。
【請求項８】

請求項１記載のシステムにおいて、前記システムは、データグラムトランスポート層セキュリティ（ＤＴＬＳ）を含むトランスポート層セキュリティ方式と統合されているシステム。
【請求項９】
請求項８記載のシステムにおいて、前記システムは、制約付きデバイスのための制約付きアプリケーションプロトコル（ＣｏＡＰ）を含むアプリケーション層プロトコルを統合していて、セッションの確立は、前記データグラムトランスポート層セキュリティ（ＤＴＬＳ）におけるセッション確立のオーバーヘッドを軽減するために前記制約付きアプリケーションプロトコル（ＣｏＡＰ）に埋め込まれているシステム。
【請求項１０】
請求項１記載のシステムにおいて、セッションをリフレッシュする鍵リフレッシュタイマーを含み、該鍵リフレッシュタイマーは、鍵リフレッシュタイマー値が最大転送カウント（MAX_RETRANSMIT_COUNT）と再送のタイムアウト時間(MAX_RETRANSMISSION_TIMEOUT)との積より大きくなったとき、各セッションをリフレッシュするシステム。
【請求項１１】
サーバーとクライアント間のデータグラム転送に関する双方向の相互認証のためのコンピュータにより実施される、システム処理コマンドを含む方法であって、
第１の乱数発生器を利用して、第１の乱数を発生するステップ；
第２の乱数発生器を利用して、第２の乱数を発生するステップ；
秘密鍵生成器を利用して、システム処理コマンドに応じて秘密鍵を生成するステップ；
システム処理コマンドに応じて、双方向認証の前であって、プロビジョニング処理の間に、前記生成された秘密鍵をサーバーとクライアントに送信するステップ；
システム処理コマンドに応じて、クライアントの固有ＩＤを含む第１のメッセージを送信するステップ；
システム処理コマンドに応じて、前記第１のメッセージを受信し、受信したクライアントＩＤを予め格納されているクライアントＩＤと照合するステップ；
前記受信したクライアントＩＤに基づいてクライアントを識別するステップ；
セッション鍵生成器を利用して、セッション鍵タイマー値の満了に基づいてセッション鍵を無効にし、満了の上で新しいセッションの確立の要求を示す固有時間制限付きセッション鍵を生成するステップ；
セッション鍵を受信し、システム処理コマンドを受けて、第１の乱数発生器によって生成された第１の乱数と、前記セッション鍵生成器により生成された前記セッション鍵を含むチャレンジコードを生成するステップ；
前記チャレンジコード生成器により生成されたチャレンジコードを受信し、システム処理コマンドに応じて、前記第１の暗号化部を利用して前記秘密鍵生成器により生成された秘密鍵で前記受信したチャレンジコードを暗号化し、暗号化したチャレンジコードをシステム処理コマンドに応じて送信するステップと；
前記暗号化されたチャレンジコードを受信し、システム処理コマンドに応じて、前記第１の復号化器を利用して、前記秘密鍵生成器により生成された秘密鍵で暗号化されたチャレンジコードを復号化し、第１の乱数とセッション鍵を取得するステップ；
前記第１の復号器からセッション鍵を受信し、リポジトリに格納するステップ；
システムの処理コマンドに応じて、前記復号化された第１の乱数とセッション鍵を受信し、該復号化された第１の乱数及び前記第２の乱数発生器により発生された第２の乱数を含むセッション鍵で暗号化された第２のメッセージを送信するステップ；
システム処理コマンドに応じて、第２のメッセージを受信し、第１の乱数と第２の乱数を、前記セッション鍵生成器により生成されたセッション鍵を用いて復号化するステップ；
システムの処理コマンドに応じて、第２のメッセージからの復号化された第１の乱数と第１の乱数発生器によって発生された第１の乱数を比較するステップ；
システム処理コマンドを受けて、前記復号化された第１の乱数が前記第１の乱数発生器

によって発生された第１の乱数と一致したとき、前記クライアントを認証するステップ；
クライアントを認証した後、システムの処理コマンドに応じて、前記セッション鍵発生器によって生成されたセッション鍵とともに受信した第２のメッセージ中の第２の乱数を暗号化し、暗号化された第２の乱数を送信するステップ；
システム処理コマンドに応じて受信し、リポジトリから受信したセッション鍵で暗号化された第２の乱数を復号化するステップ；
システム処理コマンドに応じて、前記復号化された第２の乱数と前記第２の乱数発生器によって発生された第２の乱数を比較するステップ；及び
システム処理コマンドに応じて、前記復号化された第２の乱数と前記第２の乱数発生器によって発生された第２の乱数が一致するとき、相互認証を確立するためにサーバーを認証するステップ；を含み、
前記第１の乱数発生器によって発生された第１の乱数は、第１のタイマー値を付加した第１の疑似乱数であること、
を特徴とする方法。
【請求項１２】
請求項１１に記載の方法において、前記第２の乱数を発生するステップは、システム処理コマンドに応じて、第２の乱数を発生するために、第２のタイマー値の生成と該第２のタイマー値を第２の擬似乱数に付加するステップを含む方法。
【請求項１３】
請求項１１に記載の方法において、前記秘密鍵を生成するステップは、進行中のセッションの間のみ有効な、セッションの開始時に固有鍵の生成を含む方法。
【請求項１４】
請求項１１に記載の方法において、乱数を発生するステップは、システムの処理コマンドに応じて生成する数値を含み、該数値は再現不能であって異なるセッションによって変化する方法。
【請求項１５】
請求項１１に記載の方法において、クライアントは、サーバーがクライアントの実行要求のステータスに応答しないようにサーバーと通信できる方法。
【請求項１６】
請求項１１に記載の方法は、データグラムトランスポート層セキュリティ（ＤＴＬＳ）を含むトランスポート層セキュリティ方式と統合されている方法。
【請求項１７】
請求項１６に記載の方法は、制約付きデバイスのための制約付きアプリケーションプロトコル（ＣｏＡＰ）を含むアプリケーション層プロトコルと統合されていて、セッションの確立は、前記データグラムトランスポート層セキュリティ（ＤＴＬＳ）におけるセッション確立のオーバーヘッドを軽減するために前記制約付きアプリケーションプロトコル（ＣｏＡＰ）に埋め込まれている方法。
【請求項１８】
請求項１１に記載の方法は、前記鍵リフレッシュタイマーが最大転送カウント（MAX_RETRANSMIT_COUNT）と再送タイムアウト時間（MAX_RETRANSMISSION_TIMEOUT）との積より大きくなったとき、鍵リフレッシュタイマーによって各セッションをリフレッシュするステップを含む方法。