(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第5209731号
(P5209731)
(45)発行日　平成25年6月12日(2013.6.12)　(24)登録日　平成25年3月1日(2013.3.1)

(51)Int.Cl.		F I		
G06F 21/44	(2013.01)	G06F	21/20	144A
H04L 9/32	(2006.01)	H04L	9/00	673B
G06F 12/00	(2006.01)	G06F	12/00	533J

請求項の数 5　(全 35 頁)

(21)出願番号	特願2010-532407 (P2010-532407)
(86)(22)出願日	平成20年4月9日 (2008.4.9)
(65)公表番号	特表2011-504261 (P2011-504261A)
(43)公表日	平成23年2月3日 (2011.2.3)
(86)国際出願番号	PCT/CN2008/070686
(87)国際公開番号	WO2009/059496
(87)国際公開日	平成21年5月14日 (2009.5.14)
審査請求日	平成22年6月4日 (2010.6.4)
(31)優先権主張番号	200710170309.5
(32)優先日	平成19年11月8日 (2007.11.8)
(33)優先権主張国	中国 (CN)
(31)優先権主張番号	200710195462.3
(32)優先日	平成19年11月27日 (2007.11.27)
(33)優先権主張国	中国 (CN)

(73)特許権者　504277388
▲ホア▼▲ウェイ▼技術有限公司
中華人民共和国５１８１２９広東省深▲セン▼市龍岡区坂田華為本社ビル
(74)代理人　100146835
弁理士　佐伯　義文
(74)代理人　100089037
弁理士　渡邊　隆
(74)代理人　100110364
弁理士　実広　信哉
(72)発明者　柴　▲暁▼前
中華人民共和国５１８１２９広東省深▲セン▼市龍岡区坂田華為本社ビル

最終頁に続く

(54)【発明の名称】認証方法およびクライアント

(57)【特許請求の範囲】
【請求項１】
データ同期プロトコルまたはデバイス管理プロトコルに基づく認証方法であって、
クライアントが、サーバから送信された起動メッセージを受信するステップを含み、前記起動メッセージは、起動メッセージノンス(nonce)としての前記サーバのシステム時間と、前記起動メッセージノンスを使用することによって生成されたダイジェストと、を含み、
前記クライアントが、前記起動メッセージノンスを抽出するステップと、
前記クライアントが、前記クライアントのローカル時間と前記サーバのシステム時間との間の差を計算し、前記差が閾値より小さいならば前記起動メッセージノンスが有効であると判定するステップと、
前記クライアントが、前記起動メッセージノンスが有効であると判定した後、前記起動メッセージノンスを使用してダイジェストを生成し、前記生成されたダイジェストを前記起動メッセージに含まれるダイジェストと比較することによって前記起動メッセージを認証するステップと、
前記クライアントが、前記認証に成功した後、前記起動メッセージによって指示されたサーバに、セッション識別子を伝送するセッション要求を送信するステップと、
をさらに含むことを特徴とする方法。
【請求項２】
前記起動メッセージを受信するステップの前に、

前記方法は、
前記サーバが、サーバノンスを使用して起動メッセージを生成するステップと、
前記サーバが、前記サーバノンスを使用することによって生成された起動メッセージを前記クライアントに送信するステップと、
前記サーバが、前記サーバノンスを使用することによって生成された起動メッセージの認証に失敗したと判定した後、前記起動メッセージノンスを使用して起動メッセージを生成するステップと、
をさらに含むことを特徴とする請求項１に記載の方法。
【請求項３】
前記起動メッセージノンスは前記起動メッセージのメッセージヘッダ内またはメッセージ本体内で伝送され、前記起動メッセージのメッセージヘッダおよびメッセージ本体を使用することによってダイジェストが生成されることを特徴とする請求項１または２に記載の方法。
【請求項４】
起動メッセージノンスとしてのサーバのシステム時間と、前記起動メッセージノンスを使用することによって生成されたダイジェストと、を含み、データ同期プロトコルまたはデバイス管理プロトコルに準拠する起動メッセージを受信するように構成された受信ユニットと、
前記起動メッセージノンスを抽出し、クライアントのローカル時間と前記サーバのシステム時間との間の差を計算し、前記差が閾値より小さいならば前記起動メッセージノンスが有効であると判定し、前記起動メッセージノンスが有効であると判定した後、前記起動メッセージノンスを使用してダイジェストを生成し、前記生成されたダイジェストを前記起動メッセージに含まれるダイジェストと比較することによって前記起動メッセージを認証し、前記認証に成功した後、前記起動メッセージによって指示されたサーバにセッション要求を送信するように構成された第１の認証ユニットと、
を備えることを特徴とするクライアント。
【請求項５】
前記起動メッセージを受信した後、サーバノンスを使用して前記起動メッセージを認証し、前記認証に失敗した場合、前記起動メッセージノンスを使用して前記起動メッセージを再認証するように構成された第２の認証ユニットをさらに備えることを特徴とする請求項４に記載のクライアント。
【発明の詳細な説明】
【技術分野】
【０００２】
本発明は、通信技術に関し、具体的にはデータ同期(DS)プロトコルおよびデバイス管理(DM)プロトコルに基づく認証方法、ならびにシステム、サーバ、およびクライアントに関する。
【背景技術】
【０００３】
同期マークアップ言語(SyncML)は、個人情報および企業間データを複数のプラットフォームとネットワークとの間で同期させるために開発されたプロトコルである。このSyncMLプロトコルは、関係するエンティティ間の一連の動作を定義し、そのような動作を伝送するための1組のメッセージフォーマットを定義する。このSyncMLに基づいて、OMA(オープンモバイルアライアンス)が、DSプロトコルおよびDMプロトコルを開発している。
【０００４】
DSプロトコルは、個人情報および企業間データを複数のプラットフォームとネットワークとの間で同期させることができる。このDSプロトコルは、モバイルデバイスまたはアプリケーションサーバとネットワークサーバとの間のデータを同期するため、もしくは2台のPC(パーソナルコンピュータ)間のデータを同期するために一般に利用される。
【０００５】

DMプロトコルは、管理命令データをネットワークからクライアントにダウンロードし、そのクライアントのソフトウェアおよびハードウェアを更新し、構成し、診断するためにクライアントにその管理命令を自動的に実行させる、費用効率の高い遠隔管理の解決策である。このDMはさらに、オペレータが必要とするサービス情報、およびクライアントの機能に関する情報をクライアントからサーバに転送し、そして他のサービスの動作をサポートする。

【０００６】

図１に示すように、サーバおよびクライアントを効果的に認証するために、このDSプロトコルおよびDMプロトコルに、同様のセキュリティ認証メカニズムが適用される。

【０００７】

ステップ101:サーバが、セッションを起動するために起動メッセージ(Trigger message)をクライアントに送信する。

この起動メッセージは、サーバの乱数(s_nonce)を使用することによって生成されるダイジェスト、および起動情報(TriggerInfo)を伝送する。この起動メッセージは、ショートメッセージで、または他のプッシュメッセージで伝送することができる。

s_nonceは、クライアントによって生成され、サーバが利用可能な乱数(ノンス：nonce(number used once(一度だけ使われる数)))である。

【０００８】

ステップ102:クライアントが、セッション要求をサーバに送信する。

起動メッセージを受信した後、クライアントは記憶されたs_nonceを使用してダイジェスト情報を生成し、その起動メッセージを認証する。認証に成功した場合、クライアントは、セッションを開始するためにセッション要求をサーバに送信する。

このセッション要求は、セッション識別子(Session ID)、およびそのクライアントの認証情報を伝送する。この認証情報は、クライアントnonce(c_nonce)を使用することによって生成されるダイジェストである。

c_nonceは、サーバによって生成され、クライアントが利用可能である。

このステップで、クライアントとサーバとの間にセッション接続がセットアップされる。

【０００９】

ステップ103:サーバが、認証結果および認証要求を伝送する応答を返す。

クライアントによって送信される認証情報に基づいて、サーバはそのクライアントを認証し、次いで認証結果およびサーバ認証要求を伝送する応答を返す。

より詳細には、この応答は、サーバがクライアントを認証することについての結果、セッションID、およびサーバの認証情報(すなわちs_nonceを使用することによって生成されるダイジェスト)を伝送する。

【００１０】

ステップ104:クライアントが、認証結果を伝送するメッセージをサーバに返す。

サーバによって送信される認証情報に基づいて、クライアントはそのサーバを認証し、次いで認証結果を伝送するメッセージをサーバに返す。

より詳細には、このメッセージは、クライアントがサーバを認証することについての結果および他の関連情報を伝送する。

【００１１】

サーバがクライアントの認証に失敗し、またはクライアントがサーバの認証に失敗する、例えばパスワードが正しくないもしくはnonce値が正しくない場合、サーバまたはクライアントは、チャレンジ要求を相手方に直接送信して再認証することができる。

【００１２】

起動メッセージで使用されるs_nonceが正しくないことをサーバが認識した場合、例えば、サーバが起動メッセージを幾度も送信した後、そのクライアントから通常応答を受信しない場合、そのサーバはそのs_nonceが正しくないことを知り、省略時nonce“0x00000000”を使用することにより、起動メッセージのダイジェストを生成する。s_nonceを使用

することによって生成されるダイジェストに基づいて起動メッセージを認証することに失敗した後、クライアントは省略時nonceを使用してダイジェストを生成し、その起動メッセージを再認証する。その認証に成功した場合、省略時nonceを使用してサーバおよびクライアントを認証し、次いでs_nonceおよびc_nonceが更新される。図２にその更新プロセスを示す。

【００１３】

ステップ201:サーバが、セッションを起動するために起動メッセージをクライアントに送信する。

前のs_nonceが正しくないと判定した後、サーバは省略時nonceを使用して起動メッセージを生成し、そのメッセージをクライアントに送信する。この起動メッセージは、省略時nonceを使用することによって生成されるダイジェスト、および起動情報を伝送する。

【００１４】

ステップ202:クライアントが起動メッセージの認証に失敗し、省略時nonceを使用して再認証する。

起動メッセージを受信した後、クライアントは記憶されたs_nonceを使用してその起動メッセージを認証する。何らかの理由で認証に失敗した場合、クライアントは省略時nonceを使用してその起動メッセージを再認証する。

認証に成功した場合、それはサーバが前に使用したs_nonceが正しくないことを意味し、クライアントはセッション要求をサーバに送信する。

【００１５】

ステップ203:クライアントが、セッション要求をサーバに送信する。

省略時nonceを使用することにより成功裏に認証した後、クライアントは、セッションを開始するためにセッション要求をサーバに送信する。

このセッション要求は、セッションID、および省略時nonceを使用することによって生成されるダイジェストを伝送する。

【００１６】

ステップ204:サーバが、認証結果、認証要求、およびc_nonceを更新するためのコマンドを伝送する応答を返す。

サーバは、省略時nonceを使用することによりクライアントを認証し、次いで認証結果および認証要求を伝送する応答をクライアントに返す。

より詳細には、この応答は、サーバがクライアントを認証した結果、c_nonceを更新するためのコマンド、および省略時nonceを使用することによって生成されるダイジェストを伝送する。

【００１７】

ステップ205:クライアントが、認証結果、およびs_nonceを更新するためのコマンドを伝送するメッセージをサーバに返す。

クライアントは、省略時nonceを使用することによりサーバを認証する。認証に成功した後、クライアントはc_nonceを更新し、次いで認証結果、およびs_nonceを更新するためのコマンドを伝送するメッセージをサーバに返す。

【００１８】

ステップ206:サーバが、s_nonce更新結果をクライアントに返す。

【発明の概要】

【発明が解決しようとする課題】

【００１９】

本発明を開発する過程において、本発明者は、従来技術に少なくとも以下の欠点を見出す。

s_nonceが正しくない場合、認証するために省略時nonceが使用される。省略時nonceはオープンな固定値であり、悪意あるサーバは、この省略時nonceを使用するメッセージを傍受し、そのメッセージを繰り返し送信してサーバまたはクライアントを攻撃することができる。

【００２０】
従来技術では1つのセッションで2つのnonce値、つまりs_nonceおよびc_nonceが使用され、これらのnonce値は、サーバおよびクライアントによりそれぞれ生成され、更新されるため、クライアントおよびサーバに重い管理負荷を負わせる。
【課題を解決するための手段】
【００２１】
本発明の実施形態は、クライアントとサーバとの間で行われ、DSプロトコルまたはDMプロトコルに基づく認証プロセスを最適化するための、DSプロトコルまたはDMプロトコルに基づく認証方法、システム、サーバ、およびクライアントを提供する。
【００２２】
本発明の一実施形態でのDSプロトコルまたはDMプロトコルに基づく認証方法は、
サーバが、起動メッセージノンス(nonce)を使用して起動メッセージを生成し、クライアントが起動メッセージノンスを抽出することができるように、生成された起動メッセージをクライアントに送信するステップと、
起動メッセージノンスが有効であると判定した後、起動メッセージノンスを使用してダイジェストを生成し、起動メッセージノンスを使用することによって生成された起動メッセージを認証するステップと、
認証に成功した後、起動メッセージによって指示されたサーバに、セッション識別子を伝送するセッション要求を送信するステップと、
を含む。
【００２３】
本発明の一実施形態でのDSプロトコルまたはDMプロトコルに基づく別の認証方法は、
クライアントが、更新される必要があるサーバノンスを判定するステップと、
新たなサーバノンスを生成し、新たなサーバノンスをセッション要求に追加し、サーバが新たなサーバノンスを伝送するセッション要求を受信した後、新たなサーバノンスを使用して、記憶されたサーバノンスを更新することができるように、そのセッション要求をサーバに送信するステップと、
を含む。
【００２５】
本発明の一実施形態でのDSプロトコルまたはDMプロトコルに基づく別の認証方法は、
クライアントが、省略時ノンスを使用することによって生成されたセッション要求をサーバに送信するステップと、
サーバが、セッション要求を受信した後、認証が必要であると判定した場合は省略時ノンスを使用してセッション要求を認証し、認証に成功した後にクライアントに、認証結果、ランダムノンスを使用することによって生成された認証要求、および、クライアントノンスを更新するコマンドを伝送する応答を返すステップと、
クライアントが、応答を受信し、認証が必要であると判定した場合は、省略時ノンスを使用して応答を認証し、認証に成功した後にサーバに、認証結果、およびサーバノンスを更新するコマンドを伝送する応答を返すステップと、
を含む。
【００２６】
本発明の一実施形態でのDSプロトコルまたはDMプロトコルに基づく別の認証方法は、
クライアントが、サーバによって送信され、省略時ノンス、セッション識別子、または、起動メッセージ識別子を使用することによって生成された起動メッセージを受信するステップと、
起動メッセージを受信した後、サーバノンスを使用して起動メッセージを認証し、認証に失敗した場合、省略時ノンス、セッション識別子、または、起動メッセージ識別子を使用して起動メッセージを認証し、認証に成功した後、クライアントノンスを使用してセッション要求を生成し、サーバがクライアントノンスを使用してクライアントを認証することができるように、セッション要求をサーバに送信するステップと、

を含む。
【００２８】
本発明の一実施形態で提供するサーバは、
起動メッセージノンスを使用してDSプロトコルまたはDMプロトコルに準拠する起動メッセージを生成するように構成された第１の生成ユニットと、
クライアントが起動メッセージノンスを抽出することができるように、起動メッセージノンスを使用することによって生成された起動メッセージをクライアントに送信し、起動メッセージノンスが有効であると判定した後、起動メッセージノンスを使用して、その起動メッセージノンスを使用することによって生成された起動メッセージを認証し、認証に成功した後、起動メッセージによって指示されたサーバにセッション要求を送信するように構成された送信ユニットと、
を含む。
【００２９】
本発明の一実施形態で提供するクライアントは、
起動メッセージノンスを使用することによってサーバによって生成され、DSプロトコルまたはDMプロトコルに準拠する起動メッセージを受信するように構成された受信ユニットと、
起動メッセージノンスを抽出し、起動メッセージノンスが有効であると判定した後、起動メッセージノンスを使用してダイジェストを生成し、起動メッセージノンスを使用することによって生成された起動メッセージを認証し、認証に成功した後、起動メッセージによって指示されたサーバにセッション要求を送信するように構成された第１の認証ユニットと、
を含む。
【００３１】
本発明の一実施形態で提供する別のクライアントは、
起動メッセージノンスを使用することによりサーバによって生成され、DSプロトコルまたはDMプロトコルに準拠する起動メッセージを受信するように構成された受信ユニットと、
起動メッセージを受信した後、サーバノンスを使用して起動メッセージを認証し、認証に失敗した場合、省略時ノンスを使用してその起動メッセージを認証し、認証に成功した後、クライアントノンスを使用してセッション要求を生成し、サーバがクライアントノンスを使用してそのクライアントを認証することができるように、セッション要求をサーバに送信するように構成された生成ユニットと、
を含む。
【００３２】
上記の技術的解決策は、このシステムのセキュリティを効果的に向上させる。
【００３４】
上記の技術的解決策により、クライアントとサーバとの間での認証を実施するために、サーバおよびクライアントは、セッションプロセスで従来技術のs_nonceおよびc_nonceの代わりにnonceを共有するため、システム負荷を効果的に軽減する。
【図面の簡単な説明】
【００３５】
【図１】従来技術での認証方法のフロー図である。
【図２】従来技術での、認証を行うために省略時nonceを使用し、s_nonceおよびc_nonceを更新することに関するフロー図である。
【図３】本発明の実施形態１での認証方法のフロー図である。
【図４】nonceが追加された後のメッセージフォーマットの構造を示す図である。
【図５】本発明の実施形態１において、s_nonceが正しくない場合の認証方法のフロー図である。
【図６】本発明の実施形態２での認証方法のフロー図である。

【図７】本発明の実施形態４での認証方法のフロー図である。
【図８】本発明の実施形態５での認証方法のフロー図である。
【図９】本発明の実施形態６での認証方法のフロー図である。
【図１０】本発明の実施形態７での認証方法における、新たなs_nonceを伝送する状態応答メッセージフォーマットを示す図である。
【図１１】本発明の実施形態８での認証方法のフロー図である。
【図１２】本発明の実施形態９での認証方法のフロー図である。
【図１３】本発明の実施形態１０での認証方法のフロー図である。
【図１４】本発明の実施形態１での認証システムの構造を示す図である。
【図１５】本発明の実施形態１でのクライアントの構造を示す図である。
【図１６】本発明の実施形態２でのクライアントの構造を示す図である。
【図１７】本発明の実施形態２での認証システムの構造を示す図である。
【発明を実施するための形態】
【００３６】
本発明の実施形態は、クライアントとサーバとの間で行われ、DSプロトコルまたはDMプロトコルに基づく認証プロセスを最適化するための、DSプロトコルまたはDMプロトコルに基づく認証方法、システム、サーバ、およびクライアントを提供する。
【００３７】
本明細書で言及するセッション内でのメッセージ認証に適用されるセキュリティメカニズムは、アプリケーション層セキュリティメカニズムである。
【００３８】
本発明の実施形態１での認証方法では、サーバが起動メッセージ用のnonceを生成し、このnonceはs_nonceおよびc_nonceとは異なり、起動メッセージに利用可能である。このnonceは、起動メッセージnonceと呼ぶことができる。サーバはこのnonceを使用して認証情報を生成し、この新たなnonceおよび認証情報を起動メッセージとともにクライアントに送信する。クライアントは、この新たなnonceを使用してその起動メッセージを認証する。
【００３９】
図３に示すように、本発明の実施形態１での認証方法は以下のステップを含む。
ステップ301:サーバが、起動メッセージをクライアントに送信する。このメッセージは、起動メッセージnonceを伝送する。
【００４０】
起動メッセージを送信する前に、サーバは起動メッセージnonceを生成し、このnonceを使用してダイジェストを生成し、次いでそのダイジェストを使用して起動メッセージを生成する。
【００４１】
この実施形態では、この起動メッセージnonceを使用する３種類の方法がある。
【００４２】
（１）起動メッセージを生成するとき、サーバは、自らのシステム時間(Ts)を起動メッセージnonceとして使用し、このTsを起動メッセージ内に追加する。したがって、その起動メッセージを受信した後、クライアントは、ローカル時間(Tc)をシステム時間(Ts)と比較することにより、そのnonceの有効性を判定することができる。nonceに関しては、その有効性は一般にnonceの鮮度と呼ばれる。新鮮なnonceは有効であり、新鮮でないnonceは無効である。
その起動メッセージを受信した後、クライアントはTsとTcとの間の差、つまり|Ts-Tc|を計算する。|Ts-Tc|が、事前に設定された閾値「Diff」よりも小さい場合、その起動メッセージnonceは有効であり、|Ts-Tc|が、事前に設定された閾値「Diff」より小さくない場合、その起動メッセージnonceは無効である。
閾値Diffは、概してクライアントにおいて設定され、ネットワーク状態に応じて決定される経験値とすることができる。モバイルネットワーク自体が安定しておらず、起動メッ

セージの伝送遅延を引き起こす傾向があるからである。小さすぎる閾値は、起動メッセージnonceを無効にする傾向があり、閾値が大きすぎる場合、悪意あるサーバが起動メッセージを傍受し、そのメッセージをクライアントに繰り返し続ける場合、クライアントはそれらのメッセージを有効な情報とみなし、|Ts-Tc|がその閾値の範囲に入る限りそれらのメッセージを処理する。より大きい閾値は、より攻撃を受けやすい。

【００４３】

（２）起動メッセージを生成する前に、サーバはまず特定の規則に従って起動メッセージ用のセッションIDを生成する。この規則は、現在のセッションIDから前のセッションIDを推論することを可能にする。セッションIDは、起動メッセージnonceの役割をする。サーバはこのnonceを使用してダイジェストを生成し、そのダイジェストを使用して起動メッセージを生成する。

起動メッセージを受信した後、クライアントは、その起動メッセージによって起動されようとするセッションのセッションIDを抽出し、このセッションID、サーバID、サーバパスワード、および起動メッセージの他のフィールドを使用して、そのメッセージを認証するためのダイジェストを生成する。認証に成功した後、クライアントはセッション要求を送信して、そのセッションIDに対応するセッションをセットアップする。サーバは、そのセッション要求からセッションIDを抽出してセッションを識別する。

さらに、クライアントが、その起動メッセージによって起動されようとするセッションのセッションIDを抽出した後、そのクライアントは、そのセッションIDの符号化規則に従ってそのセッションIDの鮮度を推測し、またはそのクライアントは、使用したセッションIDを記憶し、起動メッセージのセッションIDを、記憶されたセッションIDと比較して鮮度を判定することができる。

この方法では、セッションIDは、起動メッセージID(通知メッセージID)によって置換することができる。この起動メッセージIDは、クライアントによって返される起動メッセージ処理結果を、起動メッセージに対応付ける。

【００４４】

（３）起動メッセージを生成するとき、サーバは各起動メッセージを番号付けし、その番号を排他的起動メッセージnonceとして使用する。サーバはこのnonceを使用してダイジェストを生成し、そのダイジェストを使用して起動メッセージを生成する。

この番号は昇順または降順とすることができる。起動メッセージを受信した後、クライアントはそのメッセージ内で伝送されるnonceを、記憶されたnonceと比較する。番号が昇順である場合、新たなnonceの方が大きい場合、そのnonceは有効であり、そうでなければそのnonceは無効である。番号が降順である場合、新たなnonceの方が小さい場合、そのnonceは有効であり、そうでなければそのnonceは無効である。

新たなnonceが有効であると判定し、サーバを成功裏に認証した後、クライアントはその新たなnonceを記憶し、その新たなnonceは、次の起動メッセージnonceと比較するために利用することができる。

【００４５】

この方法では、悪意あるサーバが起動メッセージを傍受し、このメッセージをクライアントに繰り返し送信することによりそのクライアントを攻撃した場合、この起動メッセージによって使用されるnonceは記録されているので、すべての悪意あるメッセージが無効であると判定され、よって悪意あるサーバからの攻撃を防ぐ。

【００４６】

さらに、モバイルネットワークの不安定性のため、後に送信されるメッセージがクライアントに最初に届くことがあり、様々なセッションでサーバによって送信される起動メッセージが、変更された順序でクライアントに届き、その結果、クライアントが、有効なメッセージを誤って無効なメッセージと判定する可能性がある。

例えば、サーバが3つの異なるセッションで3つの起動メッセージを連続して送信する。この3つの起動メッセージによって使用される起動メッセージnonceはそれぞれ、30、31、および32である。しかし、モバイルネットワークの不安定性のため、クライアントはその

nonceが32である起動メッセージを最初に受信する。したがって、クライアントはこのメッセージが有効であると判定し、このnonceを記録する。残りの2つの起動メッセージがこのクライアントに届くとき、このクライアントはそれら2つのメッセージのnonceを、記録されたnonceと比較する。それら2つのメッセージのnonceが、記録されたnonceよりも小さいので、クライアントはそれらのメッセージを無効であると誤って判定する。

【００４７】

そのような問題に関して、本発明の実施形態は３つの解決策を提案する。

解決策1:クライアントがすべての起動メッセージnonce値、または最後に受信した起動メッセージnonceを記憶し、無効であると判定された起動メッセージnonceを、その記憶したnonceと比較する。そのnonceが記憶したnonceと異なる場合、クライアントはそのnonceが有効であると判定し、そのnonceを記憶する。

記憶空間が限られている場合、記憶空間が設定され、記憶されたnonce値の量が上限に達した場合、記憶された最も小さいnonceが削除される。

【００４８】

解決策2:起動メッセージnonce値が昇順で番号付けされる場合、クライアントは受信した最も大きいnonce、および現在の最大値よりも小さくかつ受信されていないnonce値のすべてまたは一部を記憶し、そのクライアントは、無効であると判定された起動メッセージnonceを、その記憶したnonce値と比較し、そのnonceが記憶したnonce値と異なる場合、そのnonceが有効であると判定し、そのnonceを記憶する。起動メッセージnonce値が降順で番号付けされる場合、クライアントは受信した最も小さいnonce、および現在の最小値よりも大きくかつ受信されていないnonce値のすべてまたは一部を記憶し、そのクライアントは、無効であると判定された起動メッセージnonceを、その記憶したnonce値と比較し、そのnonceが記憶したnonce値と異なる場合、そのnonceが有効であると判定し、そのnonceを記憶する。

例えば、初期値が1であり、番号付け方式が昇順であり、クライアントが起動メッセージnonce値、1、2、4、5、および7を順次受信すると仮定する。この場合、このクライアントは最も大きいnonceである「7」、ならびに7よりも小さくかつ受信されていないnonce値、つまり「3」および「6」を記録する。このクライアントが起動メッセージnonce「6」を受信するとき、このクライアントは「6」を最大値の「7」と比較する。6は7よりも小さいので、このnonceは無効である。次いで、このクライアントは「6」を「3」および「6」と比較し、同じ値を見つけ、その結果このクライアントは、この起動メッセージnonceが有効であると判定し、記録されていた「6」を削除する。番号付け方式が降順である場合にも判定方法は同様であり、ここではこれ以上繰り返さない。

【００４９】

解決策3:起動メッセージnonce値が昇順で番号付けされる場合、クライアントは最も大きいnonceを記憶し、その記憶した最も大きいnonceよりもnonceが小さいすべての起動メッセージを無効であるとみなす。起動メッセージnonce値が降順で番号付けされる場合、クライアントは最も小さいnonceを記憶し、その記憶した最も小さいnonceよりもnonceが大きいすべての起動メッセージを無効であるとみなす。サーバが、ある期間内にクライアントから応答を受信しない場合、そのサーバは番号付け規則に従って新たなnonceを生成し、その新たなnonceを伝送する起動メッセージを送信する。

【００５０】

上記に説明したのは、本発明の一実施形態での起動メッセージnonceを使用する方法である。

【００５１】

システム時間または起動メッセージ番号を起動メッセージnonceとして使用する場合、起動メッセージnonceは、起動メッセージのメッセージヘッダ内またはメッセージ本体内で伝送することができる。メッセージヘッダを例にとると、図４に示すように、nonceが追加されたこのメッセージフォーマットは、ダイジェスト、起動メッセージヘッダ、および起動メッセージ本体を含む。

【００５２】
起動メッセージヘッダは、バージョン、ユーザ対話方式(UI方式)、セッションイニシエータ、nonce、予備フィールド(将来使用)、セッションID、サーバ識別子の長さ(長さ識別子)、およびサーバ識別子を含む。
【００５３】
さらに、本発明の実施形態は、起動メッセージnonceを使用してダイジェストを生成する次の２つの方法を提供する。
方法1:仮にH=MD5ハッシュ関数とし、B64=Base64符号化関数とする。ダイジェストは次式で表すことができる。
ダイジェスト=H(B64(H(サーバ識別子:パスワード)):nonce:B64(H(起動)))、
ただし、サーバ識別子フィールドはサーバ識別子であり、パスワードフィールドはサーバパスワードであり、nonceフィールドは起動メッセージnonce(すなわち上述のシステム時間(Ts)、セッションID、または起動メッセージ番号)であり、起動フィールドは、起動メッセージの起動メッセージヘッダおよび起動メッセージ本体を含む。
起動メッセージを受信し、その起動メッセージ内で伝送される起動メッセージnonceが有効であると判定した後、クライアントは、そのサーバに対応するパスワードを求めてクライアント管理ツリーを検索する。クライアントは、見つかった「パスワード」、起動メッセージ内の「サーバ識別子」、nonce、および「起動」を使用してダイジェストを生成し、生成したダイジェストをメッセージ内で伝送されるダイジェストと比較する。生成したダイジェストが同じである場合は認証に成功し、そうでなければ認証に失敗する。
【００５４】
方法2:仮にH=MD5ハッシング関数とし、B64=Base64符号化関数とする。
起動メッセージnonceは、メッセージヘッダ内またはメッセージ本体内で伝送されるので、nonceは、起動メッセージの起動メッセージヘッダフィールドおよび起動メッセージ本体フィールドの一部分となる。したがって、ダイジェストを計算するには、起動メッセージヘッダフィールドおよび起動メッセージ本体フィールドしか使用する必要がない。ダイジェストは次式で表すことができる。
ダイジェスト=H(B64(H(サーバ識別子:パスワード)):B64(H(起動)))、
ただし、「サーバ識別子」フィールドはサーバ識別子であり、「パスワード」フィールドはサーバパスワードであり、「起動」フィールドは、起動メッセージの起動メッセージヘッダおよび起動メッセージ本体を含む。
起動メッセージを受信し、その起動メッセージ内で伝送された起動メッセージnonceが有効であると判定した後、クライアントは、そのサーバに対応するパスワードを求めてクライアント管理ツリーを検索する。クライアントは、見つかった「パスワード」、起動メッセージ内の「サーバ識別子」、および「起動」を使用してダイジェストを生成し、生成したダイジェストをメッセージ内で伝送されたダイジェストと比較する。生成したダイジェストが同じである場合は認証に成功し、そうでなければ認証に失敗する。
【００５５】
ステップ302:クライアントは情報が有効であると判定し、その情報を成功裏に認証し、次いでセッション要求をサーバに送信する。
起動メッセージを受信した後、クライアントは、その起動メッセージ内で伝送された起動メッセージnonceが有効であるかどうかを判定する。この判定方法については上記で説明した。起動メッセージnonceが有効であると判定した場合、クライアントは、そのサーバに対応するパスワードを求めてクライアント管理ツリーを検索する。クライアントは、見つかった「パスワード」、起動メッセージ内の「サーバ識別子」、および「起動」を使用してダイジェストを生成し、その起動メッセージを認証する。詳細な認証方法についてはステップ301で説明した。クライアントの認証方法は、ダイジェストを生成する方法によって異なる。
認証に成功した後、クライアントは、セッションを開始するためにセッション要求をサーバに送信する。

このセッション要求は、セッションID、およびc_nonceを使用することによって生成されるダイジェストを含む認証情報を伝送する。

このステップで、クライアントとサーバとの間にセッション接続がセットアップされる。

【００５６】

ステップ303:サーバが、認証結果および認証要求を伝送する応答を返す。

クライアントによって送信される認証情報に基づいて、サーバはそのクライアントを認証し、次いで認証結果および認証要求を伝送する応答をクライアントに返す。

より詳細には、この応答は、サーバがクライアントを認証した結果、セッションID、およびs_nonceを使用することによって生成されるダイジェストを含む認証情報を伝送する。

【００５７】

ステップ304:クライアントが、認証結果を伝送するメッセージをサーバに返す。

サーバによって送信される認証情報に基づいて、クライアントはそのサーバを認証し、次いで認証結果を伝送するメッセージをサーバに返す。

より詳細には、このメッセージは、クライアントがサーバを認証することについての結果および他の関連情報を伝送する。

さらに、起動メッセージnonce値が昇順で番号付けされる場合、起動メッセージが増すに従ってnonce値はより大きくなり、起動メッセージnonce値が降順で番号付けされる場合、起動メッセージが増すに従ってnonce値は0になるまで減少する。そのような場合にはnonceを調節する必要があり、例えばカウントの開始点を調節する必要がある。本発明の実施形態は、nonce値を必要な場合に調節するいくつかの方法を提供する。

【００５８】

方法1:サーバが、クライアントにおける自らのアカウントパスワードを周期的に更新する。サーバが、クライアントにおける自らのアカウントパスワードを更新するとき、サーバおよびクライアントは、nonce値を自動的にリセットすることができる。

【００５９】

方法2:nonceを調節する必要がある場合(例えば事前に設定した時間に達する、またはカウントが事前に設定した値に達した場合)、サーバはnonceをリセットするためのコマンドを発行する。このコマンドは、例えば以下のようにアラートコマンドとすることができる。

```
<Alert>
<CmdID>1</CmdID>
<Data>1227</Data><!--nonceカウントを置換する-->
</Alert>
```

nonceを調節した後、サーバは、クライアントにおける自らのアカウントパスワードを変更するためのコマンドを発行し、そうして悪意あるサーバによってメッセージが傍受され、攻撃されることを防ぐ。

【００６０】

方法3:サーバは、クライアントのクライアント管理ツリーを直接操作することができるので、サーバはクライアント管理ツリー上の自らのアカウント情報にノードを追加し、そのノードを使用して、クライアントによって受信され保持されるnonce値を記憶することができる。ノードは次のものとすることができる。

```
<X>/AppAuth/<X>/SNAAuthCount
```

その後、nonceを調節する必要がある場合(例えば事前に設定した時間に達する、またはカウントが事前に設定した値に達した場合)、サーバは置換コマンドをそのノードに発行する。コマンドの例は以下の通りである。

```
<Replace>
<CmdID>4</CmdID>
<Item>
```

<Target>
<LocURI>./DMAcc/serverA/AppAuth/1/SNAAuthCount</LocURI>
</Target>
<Data>1</Data>
</Item>
</Replace>

nonceを調節した後、サーバは、クライアントにおける自らのアカウントパスワードを変更するためのコマンドを発行し、そうして悪意あるサーバによってメッセージが傍受され、攻撃されることを防ぐ。

【００６１】

方法4:nonceを調節する必要がある場合(例えば事前に設定した時間に達する、またはカウントが事前に設定した値に達した場合)、クライアントは置換要求をサーバに送信する。クライアントがサーバから肯定応答を受信した後、両者がnonceを調節する。調節が完了次第、サーバは、クライアントにおける自らのアカウントパスワードを更新し、そうして悪意あるサーバによってメッセージが傍受され、攻撃されることを防ぐ。

【００６２】

本発明の実施形態１で提供する認証方法では、s_nonceおよびc_nonceとは異なり、起動メッセージが利用可能なnonceを提供する。新たなセッションが開始されると、サーバは、セッションを起動するための起動メッセージによって排他的に使用されることになるnonceを生成する。クライアントは、そのnonceを使用して起動メッセージを認証する。たとえサーバによって記憶されるs_nonceが正しくなくても、クライアントは依然としてセッションを開始することができる。この場合、s_nonceまたはc_nonceが正しくない場合、認証を実施するための対話により、そのs_nonceまたはc_nonceを更新することができる。

【００６３】

正しくないs_nonceを例にとると、図５に示すように、本発明の実施形態１で提供する認証方法は以下のステップを含む。

【００６４】

ステップ501:サーバが、起動メッセージをクライアントに送信する。このメッセージは起動メッセージnonceを伝送する。

送信する前に、サーバは起動メッセージnonceを生成し、このnonceを使用してダイジェストを生成し、次いでそのダイジェストを使用して起動メッセージを生成する。

【００６５】

ステップ502:クライアントは、その起動メッセージ内で伝送される起動メッセージnonceが有効であると判定し、そのメッセージを成功裏に認証し、次いでセッション要求をサーバに送信する。

起動メッセージを受信した後、クライアントは、その起動メッセージ内で伝送される起動メッセージnonceが有効であるかどうかを判定する。この判定方法については上記で説明した。起動メッセージnonceが有効であると判定した場合、クライアントは、そのサーバに対応するパスワードを求めてクライアント管理ツリーを検索する。クライアントは、見つかった「パスワード」、起動メッセージ内の「サーバ識別子」、および「起動」を使用してダイジェストを生成し、その起動メッセージを認証する。詳細な認証方法についてはステップ301で説明した。クライアントの認証方法は、ダイジェストを生成する方法によって異なる。

認証に成功した後、クライアントは、セッションを開始するためにセッション要求をサーバに送信する。

このセッション要求は、セッションID、およびc_nonceを使用することによって生成されるダイジェストを含む認証情報を伝送する。

このステップで、クライアントとサーバとの間にセッション接続がセットアップされる。

【００６６】

ステップ503:サーバが、認証結果および認証要求を伝送する応答を返す。
クライアントによって送信される認証情報に基づいて、サーバはそのクライアントを認証し、次いで認証結果および認証要求を伝送する応答をクライアントに返す。
より詳細には、この応答は、サーバがクライアントを認証した結果、セッションID、およびs_nonceを使用することによって生成されるダイジェストを含む認証情報(認証)を伝送する。

【００６７】
ステップ504:クライアントが、記憶されたs_nonceを使用してサーバを認証するが、認証に失敗する。

【００６８】
ステップ505:クライアントが、チャレンジメッセージおよびs_nonce更新メッセージをサーバに送信する。

【００６９】
ステップ506:サーバが、新たなs_nonceを使用して認証情報を生成し、認証要求を再びクライアントに送信する。
更新メッセージを受信した後、サーバが、クライアントによって指示されるものとしての記憶されたs_nonceを更新し、更新したs_nonceを使用して新たな認証要求を生成し、その更新結果および新たな認証要求をサーバに送信する。

【００７０】
ステップ507:クライアントが、認証結果を伝送するメッセージをサーバに返す。
サーバによって送信され、更新されたs_nonceを使用することによって生成される認証要求に基づいて、クライアントはそのサーバを認証し、次いで認証結果を伝送するメッセージをサーバに返す。
より詳細には、このメッセージは、クライアントがサーバを認証することについての結果および他の関連情報を伝送する。

【００７１】
本発明の実施形態１で提供する認証方法では、新たなセッションが開始されると、サーバは、セッションを起動するための起動メッセージが排他的に利用可能なnonceを生成する。本発明の実施形態２で提供する認証方法では、このセッションを起動するための起動メッセージが排他的に利用可能なnonceは、s_nonceが正しくないとサーバがみなす場合にのみ生成される。

【００７２】
図６に示すように、本発明の実施形態２で提供する認証方法は以下のステップを含む。

【００７３】
ステップ601:サーバが、起動メッセージをクライアントに送信する。このメッセージはs_nonceを伝送する。

【００７４】
ステップ602:サーバが、認証に失敗したことを知る。
サーバが、クライアントから返されるメッセージを特定の期間内に受信しない場合、そのサーバは認証が失敗したとみなす。

【００７５】
ステップ603:サーバが、新たな起動メッセージをクライアントに送信する。このメッセージは、起動メッセージnonceを伝送する。
送信する前に、サーバは起動メッセージnonceを生成し、このnonceを使用してダイジェストを生成し、次いでそのダイジェストを使用して起動メッセージを生成する。

【００７６】
ステップ604:クライアントは、その起動メッセージ内で伝送される起動メッセージnonceが有効であると判定し、そのメッセージを成功裏に認証し、次いでセッション要求をサーバに送信する。
起動メッセージを受信した後、クライアントは、その起動メッセージが起動メッセージ

nonceを使用しているかどうかを判定することにより、認証するためにs_nonceを使用するか、それとも起動メッセージnonceを使用するかを決定する。

その判定方法は次の通りである。クライアントは、その起動メッセージがnonceフィールドを含むかどうかを判定することにより、その起動メッセージが起動メッセージnonceを使用しているかどうかを判定する。つまり、起動メッセージがnonceフィールドを含む場合、その起動メッセージは起動メッセージnonceを使用している。あるいは、クライアントは、起動メッセージ内のバージョンフィールド情報を判定することにより、その起動メッセージが起動メッセージnonceを使用しているかどうかを判定する。それは、メッセージのバージョンは、そのメッセージが起動メッセージnonceを使用しているかどうかを明らかにするからである。

起動メッセージを受信した後、クライアントは、その起動メッセージ内で伝送される起動メッセージnonceが有効であるかどうかを判定する。この判定方法については上記で説明した。起動メッセージnonceが有効であると判定した場合、クライアントは、そのサーバに対応するパスワードを求めてクライアント管理ツリーを検索する。クライアントは、見つかった「パスワード」、起動メッセージ内の「サーバ識別子」、および「起動」を使用してダイジェストを生成し、その起動メッセージを認証する。詳細な認証方法についてはステップ301で説明した。クライアントの認証方法は、ダイジェストを生成する方法によって異なる。

認証に成功した後、クライアントは、セッションを開始するためにセッション要求をサーバに送信する。

このセッション要求は、セッションID、およびc_nonceを使用することによって生成されるダイジェストを含む認証情報を伝送する。

このステップで、クライアントとサーバとの間にセッション接続がセットアップされる。

【００７７】

ステップ605:サーバが、認証結果および認証要求を伝送する応答を返す。

このステップはステップ503と同様であり、ここではこれ以上詳述しない。

【００７８】

ステップ606:クライアントが、記憶されたs_nonceを使用してサーバを認証するが、認証に失敗する。

【００７９】

ステップ607:クライアントが、チャレンジおよび新たなs_nonceを伝送するメッセージをサーバに送信する。

【００８０】

ステップ608:サーバが、更新結果および新たな認証要求をクライアントに返す。

このステップはステップ506と同様であり、ここではこれ以上詳述しない。

【００８１】

ステップ609:クライアントが、認証結果を伝送するメッセージをサーバに返す。

このステップはステップ507と同様であり、ここではこれ以上詳述しない。

【００８２】

本発明の実施形態２で提供する認証方法では、サーバがs_nonceを正しくないとみなす場合、そのサーバは、セッションを起動するための起動メッセージが排他的に利用可能なnonceを生成する。クライアントが、本発明の実施形態１で提供する認証方法に従って起動メッセージを処理する。本発明の実施形態３で提供する認証方法では、クライアントは、その起動メッセージが、その起動メッセージに排他的に利用可能なnonceを使用しているかどうかを判定することにより、s_nonceを更新するかどうかを決定することができる。その起動メッセージが、その起動メッセージに排他的に利用可能なnonceを使用している場合、サーバが、セッション内で認証するために、更新されたs_nonceを直接使用できるように、クライアントはs_nonceを直接更新する。

【００８３】

本発明の第１の実施形態および第２の実施形態で提供する認証方法では、認証に失敗した場合、認証を実施するために省略時nonceは不要であるため、このシステムのセキュリティを向上させる。

【００８４】

本発明の実施形態３で提供する認証方法では、s_nonceおよびc_nonceが従来技術に基づいて更新され、サーバパスワードおよびクライアントパスワードがそれに応じて更新される。更新されたサーバパスワードは、省略時nonceを使用することによって生成される別のサーバダイジェストをもたらし、クライアントへのメッセージリプレイ攻撃を防ぐ。更新されたクライアントパスワードは、省略時nonceを使用することによって生成される別のクライアントダイジェストをもたらし、サーバへのメッセージリプレイ攻撃を防ぎ、システムのセキュリティを向上させる。

【００８５】

本発明の実施形態４で提供する認証方法では、各ステップ間の相関が強化され、システムのセキュリティを向上させる目的で、前のステップを次のステップの認証基準として使用する。図７に示すように、この認証方法は以下のステップを含む。

【００８６】

ステップ701:クライアントが、セッションを起動するための起動メッセージを受信し、そのメッセージを認証する。

【００８７】

ステップ702:クライアントが起動メッセージの認証に失敗し、省略時nonceを使用して再認証する。

【００８８】

ステップ703:クライアントがセッション要求をサーバに送信し、このセッション要求は省略時nonceを使用することによって生成される。

省略時nonceを使用することによって行われる認証が成功した場合、クライアントは、起動メッセージによって指示されるサーバにセッション要求を送信する。そのセッションが、アプリケーション層セキュリティメカニズムを使用している場合、そのメッセージは、省略時nonceを使用することによって生成される認証情報を伝送し、このプロセスはステップ704に進む。省略時nonceを使用することによって行われる認証が失敗した場合、クライアントはどんなセッションも開始することなく起動メッセージを無視し、このプロセスは終了される。

【００８９】

ステップ704:サーバが、クライアントによって送信されるセッション要求を認証する。

サーバは、このセッション要求を次の2つの方法で認証する。

【００９０】

方法1:サーバが、c_nonceを使用して認証用の認証情報を生成し、認証に成功した場合、従来技術による標準的プロセスを実行する。認証に失敗した場合、サーバは省略時nonceを使用して認証情報を生成し、再び認証を行う。その認証に成功した場合、サーバは省略時nonceを使用してサーバ認証要求を生成し、このプロセスはステップ705に進む。

【００９１】

方法2:サーバは、そのセッションがアプリケーション層セキュリティメカニズムを使用していると判定した場合、そのサーバは、省略時nonceを使用することによって生成される起動メッセージをクライアントに送信しており、その起動メッセージはセッション要求を起動するための起動メッセージであり、そのサーバは、省略時nonceを使用してそのセッション要求を認証する。認証に成功した後、このプロセスはステップ705に進む。

【００９２】

セッション要求が、起動メッセージによって起動されたかどうかを判定する方法は次の通りである。各セッション要求は、固有のセッションIDを有する。セッション要求内で伝送されるセッションIDが、起動メッセージ内で伝送されるセッションIDと比較される。このセッションIDが同じである場合、そのセッションは起動メッセージによって起動された

ことを意味する。

【００９３】

サーバが、省略時nonceを使用することによって生成される起動メッセージをクライアントに送信しているかどうか、およびその起動メッセージがセッション要求を起動するための起動メッセージであるかどうかを判定するステップは、省略時nonceを使用することによりセッション要求が成功裏に認証された後に発生することができる。この判定に成功した後、このプロセスはステップ705に進む。

このステップの目的は、次の通りである。サーバが、省略時nonceを使用することによって生成され、セッションを起動するように設計された起動メッセージをクライアントに送信していないが、クライアントによって送信され、省略時nonceを使用することによって生成されたセッション要求を受信した場合、そのメッセージは正常または安全でなく、恐らくは悪意ある第三者によって送信された不正メッセージであり、破棄可能であることを意味する。したがって、このステップはシステムのセキュリティを向上させる。

【００９４】

ステップ705:サーバが、応答メッセージをクライアントに返す。

サーバは、認証結果、認証要求、およびc_nonceを更新するためのコマンドを伝送する応答をクライアントに返す。

【００９５】

ステップ706:クライアントが、サーバによって送信される応答を認証する。

そのセッションがアプリケーション層セキュリティメカニズムを使用している場合、クライアントは省略時nonceを使用してサーバを認証する。この認証方法は次の通りである。クライアントはs_nonceを使用して認証用の認証情報を生成し、認証に成功した場合、従来技術による標準的プロセスを実行する。認証に失敗した場合、クライアントは省略時nonceを使用して認証情報を生成し、再び認証を行う。その認証に成功した場合、クライアントはc_nonceを更新し、このプロセスはステップ707に進む。

【００９６】

このステップの代替策は次の通りである。そのセッションがアプリケーション層セキュリティメカニズムを使用し、かつクライアントが、省略時nonceを使用することによって生成されるセッション要求をサーバに送信している場合、クライアントは省略時nonceを使用して、サーバによって送信される応答を認証する。認証に成功した後、クライアントはc_nonceを更新し、このプロセスはステップ707に進む。

クライアントが、省略時nonceを使用することによって生成されるセッション要求をサーバに送信しているかどうかを判定するステップは、省略時nonceを使用することによって認証が行われた後に発生することができる。この認証に成功した後、判定が行われる。その判定に成功した後、このプロセスはステップ707に進む。

【００９７】

ステップ707:クライアントが応答をサーバに返す。

クライアントは、認証結果、c_nonce更新結果、およびs_nonceを更新するためのコマンドを伝送する応答をサーバに返す。

【００９８】

ステップ708:サーバが、s_nonce更新結果をクライアントに返す。

【００９９】

上記のステップを完了した後、リプレイ攻撃を防ぐためにサーバパスワードを更新することができ、またはサーバパスワードおよびクライアントパスワードの両方が更新される。

【０１００】

上記のステップでは、省略時nonceの代わりに、セッションIDがnonceの役割をすることができ、または起動メッセージIDがnonceの役割をすることができ、そうして公開nonce(public nonce)が不変であることを回避し、より高度なセキュリティを達成する。

【０１０１】

本発明の第３の実施形態および第４の実施形態で提供する認証方法は、このシステムのセキュリティを効果的に向上させる。

【０１０２】

従来技術では、s_nonceが正しくなく、更新される必要がある場合、図２のステップ203～206に示すように、その更新を行うためにコマンドを４回やりとりする必要がある。省略時nonceが更新される前に、その省略時nonceを使用する必要があるのでリスクが高い。メッセージがモバイルネットワーク内で何度もやりとりされるため、ネットワークの負荷がより高い。

【０１０３】

クライアントによって送信されるセッション要求に新たなs_nonceを追加するための技術的解決策を、本明細書の認証方法についての実施形態で提供する。この方法で、サーバはs_nonceを直接更新し、新たなs_nonceを使用して認証し、そうしてシグナリング対話の頻度および省略時nonceを使用する頻度を減らし、システムのセキュリティを向上させ、ネットワーク負荷を軽減することができる。

【０１０４】

図８に示すように、本発明の実施形態５で提供する認証方法は以下のステップを含む。

【０１０５】

ステップ801:クライアントは、s_nonceが更新される必要があることを知る。

クライアントはs_nonceが失効したと判定し、またはサーバ内に記憶されたs_nonceがクライアント内に記憶されたs_nonceと異なることに気付き、その結果、そのs_nonceが更新される必要があることを知る。

クライアントは、サーバ内に記憶されたs_nonceとクライアント内に記憶されたs_nonceとの間の不一致を、以下の方法で発見する。

起動メッセージを受信した後、クライアントは記憶されたs_nonceを使用してその起動メッセージを認証する。何らかの理由で認証に失敗した場合、クライアントは省略時nonceを使用して、またはセッションIDもしくは起動メッセージIDをnonceとして使用してダイジェストを生成し、その起動メッセージを再認証する。

認証に成功した場合、それはサーバが前に使用したs_nonceが正しくなく、サーバ内に記憶されたs_nonceがクライアント内に記憶されたs_nonceと異なることを意味する。

【０１０６】

ステップ802:クライアントが、更新情報を伝送するセッション要求をサーバに送信する。

s_nonceが更新される必要があることを知った後、クライアントは新たなs_nonceを生成し、そのs_nonceをセッション要求に追加し、そのセッション要求をサーバに送信して、サーバにセッションを開始し、s_nonceを更新することを要求する。

このセッション要求は、セッションID、新たに生成されたs_nonce、およびc_nonceを使用することによって生成されたダイジェストを含む認証情報を伝送する。このセッション要求では、省略時nonceまたはセッションIDもしくは起動メッセージIDをnonceとして使用してダイジェストを生成することができる。

新たに生成されたs_nonceは、セッション要求(SyncML)の同期ヘッダ内または同期本体内で伝送することができる。

新たに生成されたs_nonceを同期ヘッダ内で伝送すると仮定して、その伝送方法を以下に記載する。

s_nonceを伝送するために、同期ヘッダを以下のように修正する。

SyncHdr(VerDTD, VerProto, SessionID, MsgID, Target, Source, RespURI?, NoResp?, Cred?, Chal?, Meta?)>

このs_nonceを伝送するSyncMLメッセージは、次の通りである。

<SyncML xmlns='SYNCML:SYNCML1.2'>

<SyncHdr>

...

```
<Chal>
<Meta>
<NextNonce xmlns='syncml:metinf'>LG3iZQhhdmKNHg==</NextNonce>
</Meta>
</Chal>
</SyncHdr>
<SyncBody>
...
</SyncBody>
</SyncML>
```

【０１０７】

ステップ803:サーバが、認証結果、更新結果、および認証要求を伝送する応答を返す。

セッション要求を受信した後、サーバはc_nonceを使用してそのクライアントを認証し、セッション要求内で伝送される更新されたs_nonceを使用して、記憶されたs_nonceを更新する。認証に成功し、更新が完了した後、サーバは更新されたs_nonceを使用して認証要求を生成し、認証結果、更新コマンド、および認証要求を伝送する応答をクライアントに返す。好ましくは、サーバはc_nonceを使用してセッション要求を認証する。認証に成功した後にサーバはs_nonceを更新し、そうしてサーバ内に記憶されたs_nonceとクライアント内に記憶されたs_nonceとの間の同期を保つ。

より詳細には、この応答は、サーバがクライアントを認証した結果、s_nonce更新結果、および更新されたs_nonceを使用することによって生成されるダイジェストを含む認証情報を伝送する。

【０１０８】

ステップ804:クライアントが、認証結果を伝送するメッセージをサーバに返す。

クライアントは、更新されたs_nonceを使用してサーバを認証する。認証に成功した後、クライアントは認証結果をサーバに返す。

【０１０９】

さらに、本発明の実施形態５で提供する認証方法は、シグナリング対話の頻度を減らすために、本発明の実施形態２で提供する認証方法に適用することができる。

【０１１０】

図９に示すように、本発明の実施形態６で提供する認証方法は以下のステップを含む。

【０１１１】

ステップ901:サーバが、起動メッセージをクライアントに送信する。このメッセージはs_nonceを伝送する。

【０１１２】

ステップ902:サーバが、認証に失敗したことを知る。

例えば、サーバが特定の期間内にクライアントからセッション要求を受信しない場合、そのサーバは認証が失敗したとみなす。

【０１１３】

ステップ903:サーバが、起動メッセージをクライアントに送信する。このメッセージは、起動メッセージnonceを伝送する。

送信する前に、サーバは起動メッセージnonceを生成し、このnonceを使用してダイジェストを生成し、次いでそのダイジェストを使用して起動メッセージを生成する。

【０１１４】

ステップ904:クライアントは、s_nonceが更新される必要があることを知る。

起動メッセージを受信した後、クライアントは、その起動メッセージが、その起動メッセージに排他的に利用可能なnonceを使用するかどうかを判定し、s_nonceが更新される必要があるかどうかを決定する。その結果、クライアントは、その起動メッセージが、その起動メッセージに排他的に利用可能なnonceを使用し、s_nonceが更新される必要があることを知る。

その起動メッセージが、その起動メッセージに排他的に利用可能なnonceを使用しているかどうかを判定する方法は次の通りである。クライアントは、その起動メッセージがnonceフィールドを含むかどうかを判定することにより、その起動メッセージが起動メッセージnonceを使用するかどうかを判定する。つまり、起動メッセージがnonceフィールドを含む場合、その起動メッセージは起動メッセージnonceを使用する。あるいは、クライアントは、起動メッセージ内のバージョンフィールド情報を判定することにより、その起動メッセージが起動メッセージnonceを使用するかどうかを判定する。それは、このバージョンフィールドは、そのメッセージが起動メッセージnonceを使用するかどうかを明らかにするからである。

クライアントは、その起動メッセージが起動メッセージnonce以外を使用していることを知った場合、それはs_nonceが更新される必要がないことを意味し、通常のプロセスが実行される。

【０１１５】

ステップ905:クライアントが、更新情報を伝送するセッション要求をサーバに送信する。

起動メッセージを受信し、その起動メッセージが、その起動メッセージに排他的に利用可能な起動メッセージnonceを使用していると判定した後、クライアントは、その起動メッセージ内で伝送される起動メッセージnonceが有効であるかどうかを判定する。この判定方法については上記で説明した。起動メッセージnonceが有効であると判定した場合、クライアントは、そのサーバに対応するパスワードを求めてクライアント管理ツリーを検索する。クライアントは、見つかった「パスワード」、「サーバ識別子」、および「起動」を使用してダイジェストを生成し、その起動メッセージを認証する。詳細な認証方法についてはステップ301で説明した。クライアントの認証方法は、ダイジェストを生成する方法によって異なる。

起動メッセージを成功裏に認証した後、クライアントは新たなs_nonceを生成し、そのs_nonceをセッション要求に追加し、更新情報を伝送するそのセッション要求をサーバに送信して、サーバにセッションを開始し、s_nonceを更新することを要求する。

このセッション要求は、セッションID、更新されたs_nonce、およびc_nonceを使用することによって生成されたダイジェストを含む認証情報を伝送する。

新たに生成されたs_nonceは、セッション要求の同期ヘッダ内または同期本体内で伝送することができる。

【０１１６】

ステップ906:サーバが、認証結果、更新結果、および認証要求を伝送する応答を返す。

セッション要求を受信した後、サーバはc_nonceを使用してそのクライアントを認証し、セッション要求内で伝送される更新されたs_nonceを使用して、記憶されたs_nonceを更新する。認証に成功し、更新が完了した後、サーバは更新されたs_nonceを使用して認証要求を生成し、認証結果、更新結果、および認証要求を伝送する応答をクライアントに返す。

より詳細には、この応答は、サーバがクライアントを認証した結果、s_nonce更新結果、および更新されたs_nonceを使用することによって生成されたダイジェストを含む認証情報を伝送する。

【０１１７】

ステップ907:クライアントが、認証結果を伝送するメッセージをサーバに返す。

クライアントは、更新されたs_nonceを使用してサーバを認証する。認証に成功した後、クライアントは認証結果をサーバに返す。

より詳細には、このメッセージは、サーバを認証することについての結果および他の関連情報を伝送する。

従来技術では、サーバを成功裏に認証した後、クライアントはセッションを開始しないことを決定することもある。この場合、s_nonceが失効しまたは正しくなく、更新される必要があることをクライアントが知る場合、効果的にs_nonceを更新し、またはs_nonceを

維持することは不可能である。

【０１１８】

したがって、本発明の実施形態７の認証方法は、対応する解決策を提供する。

本発明の実施形態７で提供する認証方法では、クライアントがサーバを成功裏に認証し、セッションを開始しないと決定した後、クライアントは状態応答をサーバに送信する。クライアントは、s_nonceが失効し、またはサーバ内に記憶されたs_nonceと一致しないと判定した場合、クライアントは新たなs_nonceを生成し、その新たなs_nonceを状態応答内に追加する。クライアントは、c_nonce、クライアントのユーザ名およびパスワード、ならびに応答メッセージ本体を使用してダイジェストを計算する。したがって、状態応答を受信した後、サーバは、c_nonce、クライアントのユーザ名およびパスワード、ならびに応答メッセージ本体を使用することによって計算されたダイジェストに基づいて、その情報を認証することができる。認証に成功した後、サーバは、状態応答内で伝送される新たなs_nonceに基づいて、記憶されたs_nonceを更新する。

【０１１９】

図１０に示すように、新たなs_nonceを有するこの状態応答は、ダイジェスト、通知メッセージヘッダ、および通知メッセージ本体を含む。

通知メッセージヘッダは、バージョン、ステータスコード、通知メッセージID、次のnonce、将来使用、セッションID、認証IDの長さ(長さ-認証名)、および、認証ID(認証名)を含む。

次のnonceは、新たなs_nonceである。

【０１２０】

従来技術では、s_nonceが正しくない後は、s_nonceおよびc_nonceは二度と使用されず、クライアントおよびサーバは、省略時nonceを使用して認証情報を生成する。その結果、悪意あるサーバは、任意のメッセージを傍受することにより、そのサーバまたはクライアントを攻撃することができる。

【０１２１】

s_nonceとc_nonceとは異なり、これらのnonceはサーバおよびクライアントによってそれぞれ生成され、相手方が利用することができる。したがって、これらのnonceのどちらか一方に誤りがある場合、もう一方は影響を受けない。本発明の第８の実施形態および第９の実施形態で提供する認証方法では、s_nonceに誤りがある場合に、s_nonceを個別に更新するための解決策を提供する。

【０１２２】

次の場合、s_nonceに誤りがある。それは、クライアントが、s_nonceが失効したと判定し、またはサーバ内に記憶されたs_nonceに誤りがあることを知る、例えばクライアントが、サーバ内に記憶されたs_nonceがクライアント内に記憶されたs_nonceと一致しない、または同期していないと判定した場合である。

【０１２３】

サーバ内に記憶されたs_nonceと、クライアント内に記憶されたs_nonceとの間の一致および同期を判定する方法は、次のものとすることができる。それは、サーバが、s_nonceを使用して起動メッセージを送信した後、特定の期間内にそのクライアントから応答を受信しないこと、またはサーバによって送信された起動メッセージが、省略時nonceを使用することによって生成されたダイジェストを伝送することをクライアントが知ること、またはサーバによって送信された起動メッセージ内でnonceが使用されていないことをクライアントが知ることである。

【０１２４】

s_nonceの誤りを知り、サーバを成功裏に認証した後、クライアントはセッション要求を開始する。このセッション要求では、クライアントを認証するための認証情報がc_nonceを使用することによって生成され、または基本認証方式(つまりユーザ名とパスワード)が適用され、s_nonceがさらに更新される。

【０１２５】

認証方法についての実施形態７および実施形態８に、s_nonceを更新する２つの方法をそれぞれ提供する。

【０１２６】

実施形態８では、サーバは、省略時nonceを使用して起動メッセージを生成する。図１１に示すように、この認証方法は以下のステップを含む。

【０１２７】

ステップ1101:サーバが、セッションを起動するために起動メッセージをクライアントに送信する。

前のs_nonceが正しくないと判定した後、サーバは省略時nonceを使用して起動メッセージを生成し、そのメッセージをクライアントに送信する。この起動メッセージは、省略時nonceを使用することによって生成されたダイジェスト、起動情報、および他の関連情報を伝送する。

【０１２８】

ステップ1102:クライアントが起動メッセージの認証に失敗し、省略時nonceを使用して再認証する。

起動メッセージを受信した後、クライアントは記憶されたs_nonceを使用してダイジェストを生成し、その起動メッセージを認証する。何らかの理由で認証に失敗した場合、クライアントは省略時nonceを使用してダイジェストを生成し、その起動メッセージを再認証する。

認証に成功した場合、それはサーバが前に使用したs_nonceが正しくなく、サーバ内に記憶されたs_nonceがクライアント内に記憶されたs_nonceと異なることを意味する。

【０１２９】

ステップ1103:クライアントが、更新情報を伝送するセッション要求をサーバに送信する。

省略時nonceを使用することにより、起動メッセージを成功裏に認証した後、クライアントは新たなs_nonceを生成し、そのs_nonceをセッション要求に追加し、更新情報を伝送するそのセッション要求をサーバに送信して、サーバにセッションを開始し、s_nonceを更新することを要求する。

このセッション要求は、セッションID、更新されたs_nonce、およびc_nonceを使用することによって生成されたダイジェストを含む認証情報を伝送する。

上記のステップでは、省略時nonceの代わりに、セッションIDがnonceの役割をすることができ、または起動メッセージIDがnonceの役割をすることができ、そうして公開nonceが不変であることを回避し、より高度なセキュリティを達成する。

更新されたs_nonceをセッション要求に追加する方法は、実施形態３および実施形態４での方法と基本的に同じであり、ここではこれ以上繰り返さない。

【０１３０】

ステップ1104:サーバが、認証結果、更新結果、および認証要求を伝送する応答を返す。

セッション要求を受信した後、サーバはc_nonceを使用してそのクライアントを認証し、セッション要求内で伝送された更新されたs_nonceを使用して、記憶されたs_nonceを更新する。認証に成功し、更新が完了した後、サーバは更新されたs_nonceを使用して認証要求を生成し、認証結果、更新結果、および認証要求を伝送する応答をクライアントに返す。

より詳細には、この応答は、サーバがクライアントを認証した結果、s_nonce更新結果、および更新されたs_nonceを使用することによって生成されたダイジェストを含む認証情報を伝送する。

【０１３１】

ステップ1105:クライアントが、認証結果を伝送するメッセージをサーバに返す。

クライアントは、更新されたs_nonceを使用してサーバを認証する。認証に成功した後、クライアントは認証結果をサーバに返す。

【０１３２】
実施形態９では、サーバは省略時nonceを使用して起動メッセージを生成するが、セッション要求は更新情報を運ばない。図１２に示すように、この認証方法は以下のステップを含む。
【０１３３】
ステップ1201:サーバが、セッションを起動するために起動メッセージをクライアントに送信する。
前のs_nonceが正しくないと判定した後、サーバは省略時nonceを使用して起動メッセージを生成し、そのメッセージをクライアントに送信する。この起動メッセージは、省略時nonceを使用することによって生成されたダイジェスト、起動情報、および他の関連情報を伝送する。
【０１３４】
ステップ1202:クライアントが起動メッセージの認証に失敗し、省略時nonceを使用して再認証する。
起動メッセージを受信した後、クライアントは記憶されたs_nonceを使用してダイジェストを生成し、その起動メッセージを認証する。何らかの理由で認証に失敗した場合、クライアントは省略時nonceを使用してダイジェストを生成し、その起動メッセージを再認証する。
認証に成功した場合、それはサーバが前に使用したs_nonceが正しくなく、サーバ内に記憶されたs_nonceがクライアント内に記憶されたs_nonceと異なることを意味する。
【０１３５】
ステップ1203:クライアントがその情報を成功裏に認証し、次いでセッション要求をサーバに送信する。
認証に成功した後、クライアントは、セッションを開始するためにセッション要求をサーバに送信する。
このセッション要求は、セッションID、およびc_nonceを使用することによって生成されたダイジェストを含む認証情報を伝送する。
このステップで、クライアントとサーバとの間にセッション接続がセットアップされる。
【０１３６】
ステップ1204:サーバが、認証結果および認証要求を伝送する応答を返す。
クライアントによって送信された認証情報に基づいて、サーバはそのクライアントを認証する。認証に成功した後、サーバは省略時nonceを使用して認証情報を生成し、次いで認証結果および認証要求を伝送する応答をクライアントに返す。
より詳細には、この応答は、サーバがクライアントを認証した結果、セッションID、および省略時nonceを使用することによって生成されるダイジェストを含む認証情報(認証)を伝送する。
【０１３７】
ステップ1205:クライアントが、更新コマンドおよび認証結果をサーバに返す。
クライアントは、省略時nonceを使用することによりサーバを認証する。認証に成功した後、クライアントは新たなs_nonceを生成し、s_nonceを更新するためのコマンド、およびサーバを認証した結果をそのサーバに送信する。
【０１３８】
ステップ1206:サーバが、更新結果をクライアントに返す。
更新メッセージを受信した後、サーバが、クライアントによって指示されたものとしての記憶されたs_nonceを更新し、更新結果をクライアントに返す。
【０１３９】
上記のステップでは、省略時nonceの代わりに、セッションIDがnonceの役割をすることができ、または起動メッセージIDがnonceの役割をすることができ、そうして公開nonceが不変であることを回避し、より高度なセキュリティを達成する。

この場合、システムの信頼性を向上させるために、サーバパスワードを更新することができる。

【０１４０】

別の実施形態では、サーバは省略時nonceを使用して起動メッセージを生成するが、セッション要求は省略時nonceを伝送しない。この認証方法は以下のステップを含む。

【０１４１】

前のs_nonceが正しくないと判定した後、サーバは、省略時nonceを使用して、またはセッションIDもしくは起動メッセージIDをnonceとして使用して起動メッセージを生成し、そのメッセージをクライアントに送信する。この起動メッセージは、省略時nonceまたはセッションIDもしくは起動メッセージIDを使用することによって生成されたダイジェスト、起動情報、および他の関連情報を伝送する。

【０１４２】

起動メッセージを受信した後、クライアントは記憶されたs_nonceを使用してダイジェストを生成し、その起動メッセージを認証する。何らかの理由で認証に失敗した場合、クライアントは省略時nonceを使用して、またはセッションIDもしくは起動メッセージIDをnonceとして使用してダイジェストを生成し、その起動メッセージを再認証する。その認証に成功した場合、クライアントは、セッションを開始するためにセッション要求をサーバに送信する。このセッション要求は、セッションID、およびc_nonceを使用することによって生成されたダイジェストを含む認証情報を伝送する。

【０１４３】

サーバはc_nonceを使用してセッション要求を認証する。認証に失敗した場合、サーバは、c_nonceを更新し、再認証を要求するためにチャレンジメッセージを送信する。認証に成功した後、サーバは、s_nonceを使用することによって生成された認証要求を送信し、クライアントはs_nonceを使用して認証する。その認証に失敗した場合、クライアントは、s_nonceを更新し、再認証を要求するためにチャレンジメッセージを送信する。認証に成功した後、クライアントは結果を返す。

【０１４４】

さらに、上記のステップでは、クライアントは更新されたs_nonceをセッション要求に追加し、その要求をサーバに送信することができる。サーバによって送信された認証要求は、その新たなs_nonceを使用する。

【０１４５】

上述のように、s_nonceに誤りがある場合、s_nonceのみが更新され、c_nonceは更新されない。このシステムがs_nonceの誤りを処理するとき、たとえこのシステムが、省略時nonceを使用して、またはセッションIDもしくは起動メッセージIDをnonceとして使用して認証しても、c_nonceは更新する必要がないので、クライアントはc_nonceを使用してセッション要求を生成し、そうして省略時nonceを使用する頻度、またはセッションIDもしくは起動メッセージIDをnonceとして使用する頻度を減らし、システムのセキュリティを向上させることができる。

【０１４６】

従来技術では、セッション内で使用されるs_nonceおよびc_nonceは、クライアントおよびサーバによってそれぞれ生成され、更新されるため、クライアントおよびサーバにより重い管理負荷を負わせる。

【０１４７】

本発明の実施形態１０では、従来技術のs_nonceおよびc_nonceを置換し、クライアントとサーバとの間の認証を実施するために、１つのセッション内で１つのnonceを使用する。同一のセッションが、トランスポート層セキュリティメカニズムまたはアプリケーション層セキュリティメカニズムによって保護されており、したがって、クライアントとサーバとの間の認証を実施するために同一のnonceを使用することができる。

そのnonceは、サーバまたはクライアントが生成することができる。サーバがそのnonceを生成すると仮定して、本発明の実施形態１０の認証方法を以下に詳しく述べる。

【０１４８】
実施形態１０では、以下に説明するように、nonceを更新する２つの方法を提供する。
【０１４９】
方法1:サーバがnonceを更新する。
まず、サーバが、nonce更新コマンド(NextNonce)を発行する。このNextNonceコマンドは、新たなnonceを伝送する。
NextNonceコマンドを受信した後、クライアントは、NextNonceコマンド内で伝送される新たなnonceを使用して、記憶されたnonceを更新する。
【０１５０】
この更新コマンドは、認証メッセージ内で伝送することができ、つまりそのメッセージは更新コマンドおよびサーバ認証情報を伝送する。この更新コマンドは、別の管理メッセージ内で伝送することもでき、つまりそのメッセージはサーバ認証情報を伝送しない。更新コマンドを認証メッセージ内で伝送した場合、クライアントがそのメッセージを受信した後、そのクライアントはNextNonceコマンドに基づいてnonceを更新し、次いでその更新したnonceを使用してダイジェストを生成し、情報を認証する。その認証は成功する。この場合、悪意あるサーバがこのメッセージを傍受した場合、その悪意あるサーバはクライアントに対していつでもリプレイ攻撃を開始することができる。そのような攻撃を防ぐために、このNextNonceコマンドを認証メッセージ内で伝送した場合、クライアントは、更新されていないnonceを使用することによって生成されるダイジェストを使用してその情報を認証する。メッセージを受信した後、クライアントは、更新されていないnonceを使用することによって生成されるダイジェストを使用して最初に認証する。認証に成功した後、クライアントは、記憶されたnonceをNextNonceコマンドに基づいて更新する。別の管理メッセージがnonce更新コマンドを伝送した場合、新たなnonceを伝送するメッセージと、認証情報を伝送するもう１つのメッセージとが別々に相手方に送信されるので、リプレイ攻撃のリスクは存在しない。
【０１５１】
図１３に示すように、応答がNextNonceコマンド内で伝送されると仮定すると、そのプロセスは以下の通りである。
【０１５２】
ステップ1301:サーバが、セッションを起動するために起動メッセージをクライアントに送信する。
この起動メッセージは、共有nonceを使用することによって生成されたダイジェスト、および起動情報を伝送する。
共有nonceは、サーバによって生成され、サーバおよびクライアントが利用可能である。
実際には、このステップの共有nonceは、起動メッセージnonceまたは省略時nonceとすることができる。ある場合は、サーバは、nonceを使用しないが、サーバIDおよびパスワードを使用して、セッションを起動するための起動メッセージを生成し、クライアントはサーバIDおよびパスワードを使用して、その起動メッセージを認証するためのダイジェストを生成することができる。
【０１５３】
ステップ1302:クライアントが、セッション要求をサーバに送信する。
起動メッセージを受信した後、クライアントは記憶されたs_nonceを使用してダイジェストを生成し、その起動メッセージを認証する。認証に成功した場合、クライアントは、セッションを開始するためにセッション要求をサーバに送信する。
このセッション要求は、セッションID、および共有nonceを使用することによって生成されたダイジェストを含む認証情報を伝送する。
このステップで、クライアントとサーバとの間にセッション接続がセットアップされる。
【０１５４】

ステップ1303:サーバが、認証結果および認証要求を伝送する応答を返し、その応答はNextNonceコマンドを伝送する。

クライアントによって送信された認証情報に基づいて、サーバはそのクライアントを認証する。認証に成功した後、共有nonceが更新される必要があることをサーバが知った場合、そのサーバは新たな共有nonceを生成し、次いで認証結果および認証要求を伝送する応答をクライアントに返す。この応答は、NextNonceコマンドを伝送する。

より詳細には、この応答は、サーバがクライアントを認証した結果、セッションID、更新されていないnonceを使用することによって生成されたダイジェストを含む認証情報、および新たなnonceを含むNextNonceコマンドを伝送する。

【０１５５】

ステップ1304:応答を受信した後、クライアントは、更新されていないnonceを使用してそのメッセージを認証する。

【０１５６】

ステップ1305:認証に成功し、クライアントは、NextNonceコマンド内で伝送された新たなnonceを使用して、そのNextNonceコマンドによって指示されたものとしての記憶された共有nonceを更新する。

【０１５７】

ステップ1306:クライアントが、認証結果および更新結果を伝送するメッセージをサーバに返す。

より詳細には、このメッセージは、クライアントがサーバを認証した結果、共有nonceを更新した結果、および他の関連情報を伝送する。

【０１５８】

サーバおよびクライアントは、共有nonceの有効期間を別々に定義することができる。サーバが、共有nonceが有効であると判定したとき、その共有nonceの有効期間はクライアントにとっては切れている可能性がある。したがって、クライアントにとってのその共有nonceの有効性を保つために、本発明の実施形態１０は、クライアントが共有nonceを更新するようにサーバに要求する技術的解決策を提供する。

【０１５９】

クライアントは、DMコマンドのうちのアラートコマンドを使用して、サーバに共有nonceを更新するように要求することができる。このコマンドをサーバに理解させるため、アラートタイプがコマンド内に追加される。このアラートタイプは、サーバにnonceを更新するように要求する指示である。

【０１６０】

nonceが更新される必要があるとクライアントが認識した場合、そのクライアントは、共有nonceを更新する要求を、このアラートコマンドによりサーバに送信する。この要求は、認証メッセージ内で、または別の管理メッセージ内で伝送することができる。この要求を受信した後、サーバは、特定の条件に基づいてnonceを更新するかどうかを決定する。

【０１６１】

このアラートタイプは、org.openmobilealliance.NextNonceとして定義することができる。

【０１６２】

以下に示すのは、このアラートタイプのメッセージのインスタンスである。

```
<Alert>
<CmdID>2</CmdID>
<Data>1226</Data><!--汎用アラート-->
<Item>
  <Meta>
    <Type xmlns="syncml:metinf">
org.openmobilealliance.NextNonce
```

```
      </Type>
    </Meta>
  </Data>
  </Item>
  </Alert>
```

【０１６３】

クライアントがnonceを更新する方法は、サーバがnonceを更新する方法と同様であり、ここではこれ以上繰り返さない。

【０１６４】

方法2:サーバおよびクライアントが、nonceを共同で更新する。

【０１６５】

サーバおよびクライアントはどちらも、共有nonceが更新される必要があると判定した場合、更新用の新たなnonceを生成することができる。

【０１６６】

nonceは、NextNonceコマンドによって更新することができる。以下に示すのは、更新のインスタンスである。

```
  <Chal>
  <Meta>
  <NextNonce xmlns='syncml:metinf'>LG3iZQhhdmKNHg==</NextNonce>
  </Meta>
  </Chal>
```

【０１６７】

NextNonceコマンドは、セッションプロセスのメッセージ内で伝送することができる。例えば、クライアントはNextNonceコマンドをセッション要求に追加し、そのセッション要求をサーバに送信して、そのサーバに共有nonceを更新するように要求し、またはNextNonceコマンドを別のメッセージ内に追加することができる。

【０１６８】

サーバまたはクライアントが共有nonceを更新するかどうかを問わず、認証メッセージ内で更新コマンドが伝送される場合、相手方がそのメッセージを受信した後、その相手方が最初にnonceを更新し、次いでその更新したnonceを使用してダイジェストを生成し、情報を認証する。その認証は成功する。この場合、悪意あるサーバがこのメッセージを傍受した場合、その悪意あるサーバはクライアントに対していつでもリプレイ攻撃を開始することができる。そのような攻撃を防ぐために、このNextNonceコマンドを認証メッセージ内で伝送した場合、相手方は、更新されていないnonceを使用することによって生成されるダイジェストを使用してその情報を認証することができる。認証に成功した後、その相手方は、記憶されたnonceをNextNonceコマンドに基づいて更新する。別の管理メッセージがnonce更新コマンドを伝送した場合、新たなnonceを伝送するメッセージと、認証情報を伝送するもう1つのメッセージとが別々に相手方に送信されるので、攻撃のリスクは存在しない。

【０１６９】

本発明の実施形態１０では、サーバおよびクライアントが共有nonceを使用して認証する。この場合、共有nonceに誤りがある場合、その誤りは上述の方法のうちのいずれかにより処理することができる。本発明の実施形態５のステップ803では、メッセージは新たなnonce、その新たなnonceを使用することによって生成されたダイジェストを伝送する。この場合、悪意あるサーバがこのメッセージを傍受した場合、その悪意あるサーバはこのメッセージをサーバまたはクライアントに繰り返し送信することにより、リプレイ攻撃を開始することができる。サーバまたはクライアントはこのメッセージを識別することができず、このメッセージが有効であると信じ、対応する動作を実行する。そのような攻撃を防ぐために、更新されていないnonceを使用してダイジェストを生成することができる。この方法で、メッセージを受信した後、サーバは更新されていないnonceを使用してダイ

ジェストを計算し、そのメッセージの送信側すなわちクライアントを認証し、次いで記憶されたnonceをNextNonceコマンドに基づいて更新する。

【０１７０】

本発明の実施形態１０は、システム負荷を効果的に軽減する。

【０１７１】

図１４に示すように、本発明の実施形態１で提供する認証システムは、

起動メッセージnonceを使用して起動メッセージを生成し、その生成した起動メッセージを送信するように構成されたサーバ1410と、

起動メッセージnonceを使用することによって生成された起動メッセージを受信し、その起動メッセージnonceを使用してその生成された起動メッセージを認証する、つまりその起動メッセージの有効性を検証するように構成されたクライアント1420と

を含む。

【０１７２】

サーバ1410は、

起動メッセージnonceを使用することによって起動メッセージを生成するように構成された第１の生成ユニット1412と、

起動メッセージnonceを使用することによって生成された起動メッセージを送信するように構成された送信ユニット1411と、

サーバnonceを使用することによって起動メッセージを生成し、その生成した起動メッセージをクライアントに送信するように構成された第２の生成ユニット1417と、

サーバnonceを使用することによって生成された起動メッセージの認証に失敗したと判定した後、第１の生成ユニット1412を制御して、起動メッセージnonceを使用して起動メッセージを生成するように構成された判定ユニット1413と、

第１の生成ユニット1412が、起動メッセージnonceを使用して起動メッセージを生成するときに、サーバのシステム時間を求め、そのシステム時間を、起動メッセージnonceを使用することによって生成された起動メッセージ内に追加するように構成された時間ユニット1414と、

起動メッセージnonceを使用することにより、第１の生成ユニット1412によって生成された起動メッセージを番号付けし、その番号を起動メッセージnonceとして使用するように構成された符号化ユニット1415と、

符号化ユニット1415によって生成された起動メッセージnonceを、必要な場合にリセットするように構成されたnonceリセットユニット1416と、

起動メッセージによって起動されたセッションのセッションIDを起動メッセージnonceとして使用し、そのため、クライアントが、起動メッセージを受信した後、起動メッセージnonceを使用することによってその起動メッセージを認証し、認証に成功した後、そのセッションIDに対応するセッションをセットアップすることを要求するセッション要求を送信する、ように構成されたセッションID・nonceユニット1418と

をさらに含む。

【０１７３】

符号化ユニット1415は、

起動メッセージnonceを使用することによって生成された起動メッセージを、昇順で番号付けするように構成された昇順番号付けユニット14151と、

起動メッセージnonceを使用することによって生成された起動メッセージを、降順で番号付けするように構成された降順番号付けユニット14152と

をさらに含む。

【０１７４】

クライアント1420は、

起動メッセージnonceを使用することによって生成された起動メッセージを受信するように構成された受信ユニット1421と、

起動メッセージnonceを使用して、その起動メッセージnonceを使用することによって生

成された起動メッセージを認証する、つまりその起動メッセージnonceを使用することによって生成された起動メッセージの有効性を検証するように構成された第１の認証ユニット1422と、

起動メッセージを受信した後、サーバnonceを使用してその起動メッセージを認証し、認証に失敗した場合、起動メッセージnonceを使用してその起動メッセージを再認証するように構成された第２の認証ユニット1425と、

受信ユニット1421が、起動メッセージnonceを使用することによって生成された起動メッセージを受信するときに、クライアントのローカル時間を求め、そのローカル時間とシステム時間との間の差の絶対値を、事前に設定した閾値と比較し、その絶対値が事前に設定した閾値よりも小さい場合はその起動メッセージnonceが有効であると判定し、第１の認証ユニット1422を制御して、起動メッセージnonceを使用することによって生成された起動メッセージを認証するために、その起動メッセージnonceを使用するように構成された第１の有効性判定ユニット1423と、

起動メッセージnonceを使用することによって生成された起動メッセージを受信ユニット1421が受信した後、記憶された起動メッセージnonceに基づいて、その起動メッセージ内で伝送される起動メッセージnonceが有効であると判定した場合、その起動メッセージ内で伝送される起動メッセージnonceを記憶し、第１の認証ユニット1422を制御して、起動メッセージnonceを使用することによって生成された起動メッセージを認証するために、その起動メッセージnonceを使用するように構成された第２の有効性判定ユニット1424と、

起動メッセージによって起動されるセッションのセッションIDを起動メッセージnonceとして使用し、起動メッセージnonceを使用することによって起動メッセージを認証し、認証に成功した後、そのセッションIDに対応するセッションをセットアップすることを要求するセッション要求を送信するように構成されたセッションID・nonceユニット1428と

をさらに含む。

【０１７５】

第２の有効性判定ユニット1424は、

クライアント内に記憶された起動メッセージnonceと、起動メッセージ内で伝送された起動メッセージnonceとを比較し、起動メッセージ内で伝送された起動メッセージnonceが、クライアント内に記憶された最も大きい起動メッセージnonceよりも大きい場合、またはクライアントによって受信され、記憶された起動メッセージnonce値が、起動メッセージ内で伝送された起動メッセージnonceを含まない場合、またはクライアントによって受信されていないnonce値が、起動メッセージ内で伝送された起動メッセージnonceを含む場合に、起動メッセージ内で伝送された起動メッセージnonceが有効であると判定するように構成された第１の番号付け判定ユニット14241と、

クライアント内に記憶された起動メッセージnonceと、起動メッセージ内で伝送された起動メッセージnonceとを比較し、起動メッセージ内で伝送された起動メッセージnonceが、クライアント内に記憶された最も小さい起動メッセージnonceよりも小さい場合、またはクライアントによって受信され、記憶された起動メッセージnonce値が、起動メッセージ内で伝送された起動メッセージnonceを含まない場合、またはクライアントによって受信されていないnonce値が、起動メッセージ内で伝送された起動メッセージnonceを含む場合に、起動メッセージ内で伝送された起動メッセージnonceが有効であると判定するように構成された第２の番号付け判定ユニット14242と

を含む。

【０１７６】

実施形態１の認証システムの動作方式は、本発明の実施形態１および実施形態２の認証方法の動作方式と同様であり、ここではこれ以上繰り返さない。

【０１７７】

本発明の実施形態１で提供する認証システムにより、認証に失敗した場合、認証を実施するために省略時nonceは不要であるため、このシステムのセキュリティを向上させる。

【０１７８】
本発明の実施形態１で提供するサーバは、本発明の実施形態１で提供する認証システム内のサーバと基本的に同じであり、ここではこれ以上繰り返さない。
【０１７９】
図１５に示すように、本発明の実施形態１で提供するクライアントは、
サーバによって送信される起動メッセージを受信するように構成された受信ユニット1501と、
起動メッセージを受信した後、サーバnonceが更新される必要があると判定した場合に新たなサーバnonceを生成し、その新たなサーバnonceをセッション要求に追加し、サーバがその新たなサーバnonceを伝送するセッション要求を受信した後、その新たなサーバnonceを使用して、記憶されたサーバnonceを更新することができるように、そのセッション要求をサーバに送信するように構成された第１の生成ユニット1512と、
起動メッセージを受信した後、セッションを開始しないと決定し、サーバnonceが更新される必要があると判定した場合に新たなサーバnonceを生成し、その新たなサーバnonceを状態応答内に追加し、サーバがその新たなサーバnonceを伝送する状態応答を受信した後、その新たなサーバnonceを使用して、記憶されたサーバnonceを更新することができるように、その状態応答をサーバに送信するように構成された第２の生成ユニット1503と
を含む。
【０１８０】
実施形態１で提供するクライアントの動作方式は、本発明の第４、第５、および第６の実施形態で提供する認証方法におけるクライアントの動作方式と同様であり、ここではこれ以上繰り返さない。
【０１８１】
本発明の実施形態１で提供するクライアントにより、s_nonceが更新される必要がある場合、セッション要求は更新コマンドを直接伝送し、そしてシグナリング対話の頻度を減らし、システム負荷を軽減し、認証を行うために省略時nonceを使用する頻度を減らし、システムのセキュリティを向上させる。
【０１８２】
図１６に示すように、本発明の実施形態２で提供するクライアント1600は、
サーバによって送信される起動メッセージを受信するように構成された受信ユニット1601と、
起動メッセージを受信した後、サーバnonceを使用してその起動メッセージを認証し、認証に失敗した場合、省略時nonceを使用してその起動メッセージを認証し、認証に成功した後、クライアントnonceを使用してセッション要求を生成し、サーバがそのクライアントnonceを使用してそのクライアントを認証することができるように、そのセッション要求をサーバに送信するように構成された生成ユニット1602と、
サーバnonceを新たなサーバnonceで更新した後、サーバパスワードおよびクライアントパスワードを変更するように構成されたパスワード変更ユニット1603と
を含む。
【０１８３】
実施形態２で提供するクライアントの動作方式は、本発明の第７の実施形態および第８の実施形態で提供する認証方法におけるクライアントの動作方式と同様であり、ここではこれ以上繰り返さない。
【０１８４】
本発明の実施形態２で提供するクライアントにより、s_nonceが更新される必要がある場合、s_nonceのみが更新され、c_nonceは更新されない。たとえこのシステムがs_nonceの誤りを処理するとき、認証のために省略時nonceを使用しても、c_nonceは更新する必要がないので、クライアントはc_nonceを使用してセッション要求を生成し、そうして省略時nonceを使用する頻度を減らし、システムのセキュリティを向上させることができる。
【０１８５】

図１７に示すように、本発明の実施形態２で提供する認証システムは、サーバ1710およびクライアント1720を含む。

【０１８６】

サーバ1710は、

サーバおよびクライアントによって共有されるnonceを使用して起動メッセージを生成し、その起動メッセージを受信した後、クライアントがその共有nonceを使用してその起動メッセージを認証することができるように、その起動メッセージをクライアントに送信するように構成された起動ユニット1711と、

共有nonceを使用することによって生成されたセッション要求を、クライアントから受信するように構成された受信ユニット1712と、

共有nonceを使用することによってセッション要求を認証するように構成された認証ユニット1713と、

セッション要求が成功裏に認証された後、共有nonceを使用して応答を生成し、その応答を受信した後、クライアントがその共有nonceを使用してその応答を認証することができるように、その応答をクライアントに送信するように構成された生成ユニット1714と、

共有nonceを生成し、その共有nonceが更新される必要がある場合に新たな共有nonceを生成し、nonce更新メッセージを受信した後、クライアントがその新たな共有nonceを使用して共有nonceを更新することができるように、その新たな共有nonceを伝送するnonce更新メッセージをクライアントに送信するように構成された更新ユニット1715と、

共有nonceが更新される必要があると判定した場合に、クライアントが、nonce更新要求を受信し、nonceを更新することおよび新たな共有nonceを伝送するnonce更新メッセージを送信することを決定した後、新たな共有nonceを生成することができるように、nonce更新要求をクライアントに送信するように構成された要求ユニット1716と

をさらに含む。

【０１８７】

クライアント1720は、

サーバによって送信され、サーバおよびクライアントによって共有されるnonceを使用することによって生成される起動メッセージを受信するように構成された受信ユニット1721と、

起動メッセージを受信した後、共有nonceを使用することによってその起動メッセージを認証するように構成された第１の認証ユニット1722と、

認証に成功した後、共有nonceを使用してセッション要求を生成し、サーバが、そのセッション要求を受信した後、共有nonceを使用してそのセッション要求を認証する、つまりそのセッション要求の有効性を検証することができるように、そのセッション要求をサーバに送信するように構成された生成ユニット1723と、

共有nonceを使用することによりサーバによって生成された応答を受信した後、その共有nonceを使用してその応答を認証するように構成された第２の認証ユニット1724と、

共有nonceを生成し、その共有nonceが更新される必要がある場合に新たな共有nonceを生成し、サーバが、nonce更新メッセージを受信した後、その新たな共有nonceを使用して共有nonceを更新することができるように、その新たな共有nonceを伝送するnonce更新メッセージをサーバに送信するように構成された更新ユニット1725と、

共有nonceが更新される必要があると判定した場合に、サーバが、nonce更新要求を受信し、nonceを更新することおよび新たな共有nonceを伝送するnonce更新メッセージを送信することを決定した後、新たな共有nonceを生成することができるように、nonce更新要求をサーバに送信するように構成された要求ユニット1726と

をさらに含む。

【０１８８】

実施形態２の認証システムの動作方式は、本発明の実施形態９の認証方法の動作方式と同様であり、ここではこれ以上繰り返さない。

【０１８９】

本発明の実施形態２で提供する認証システムにより、クライアントとサーバとの間での認証を実施するために、サーバおよびクライアントは、セッションプロセスで従来技術のs_nonceおよびc_nonceの代わりにnonceを共有するため、システム負荷を効果的に軽減する。

【０１９０】

本発明の実施形態２で提供するサーバ、および本発明の実施形態３で提供するクライアントは、本発明の実施形態２で提供する認証システム内のサーバおよびクライアントと基本的に同じであり、ここではこれ以上繰り返さない。

【０１９１】

上記の実施形態のステップのすべてまたは一部を、コンピュータプログラムによって命令されるハードウェアによって実施できることを当業者なら理解することができる。そのプログラムは、コンピュータ読み取り可能な記憶媒体に記憶することができる。その記憶媒体は、ROM(読出し専用メモリ)、磁気ディスク、またはCD(コンパクトディスク)とすることができる。

【０１９２】

上記で詳しく説明したのは、本発明による、DSプロトコルまたはDMプロトコルに基づく認証方法、システム、サーバ、およびクライアントである。本発明をいくつかの例示的実施形態によって説明したが、本発明はそのような実施形態に限定されない。本発明の範囲から逸脱することなく、当業者が本発明に修正および改変を加えることが可能であることは明らかである。それらの修正形態および改変形態が、特許請求の範囲またはその等価物によって定義する保護範囲に含まれる条件で、本発明はそれらの形態を対象として含むものとする。

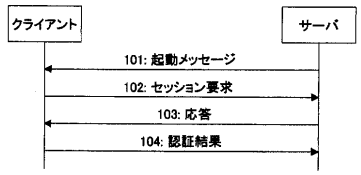
【符号の説明】

【０１９３】

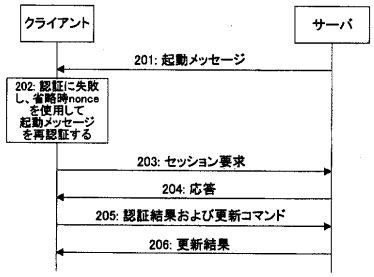
1410　サーバ
1411　送信ユニット
1412　第１の生成ユニット
1413　判定ユニット
1414　時間ユニット
1415　番号付けユニット
14151　昇順番号付けユニット
14152　降順番号付けユニット
1416　nonceリセットユニット
1417　第２の生成ユニット
1418　セッションID・nonceユニット
1420　クライアント
1421　受信ユニット
1422　第１の認証ユニット
1423　第１の有効性判定ユニット
1424　第２の有効性判定ユニット
14241　第１の番号付け判定ユニット
14242　第２の番号付け判定ユニット
1425　第２の認証ユニット
1428　セッションID・nonceユニット
1501　受信ユニット
1502　第１の生成ユニット
1503　第２の生成ユニット
1600　クライアント
1601　受信ユニット
1602　生成ユニット

1603　パスワード変更ユニット
1710　サーバ
1711　起動ユニット
1712　受信ユニット
1713　認証ユニット
1714　生成ユニット
1715　更新ユニット
1716　要求ユニット
1720　クライアント
1721　受信ユニット
1722　第１の認証ユニット
1723　生成ユニット
1724　第２の認証ユニット
1725　更新ユニット
1726　要求ユニット

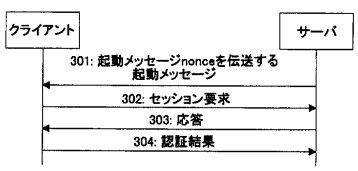
【図１】



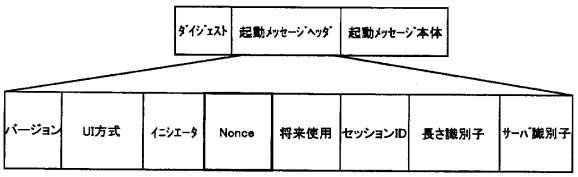
【図２】



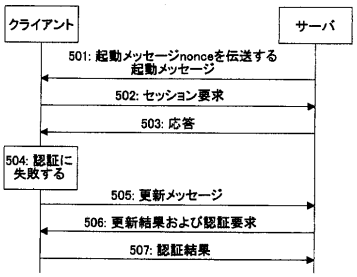
【図３】



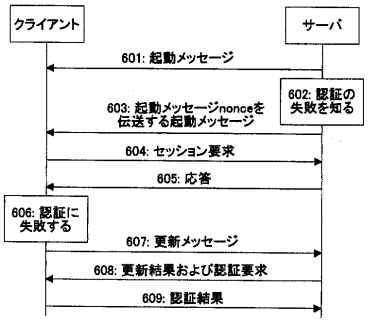
【図４】



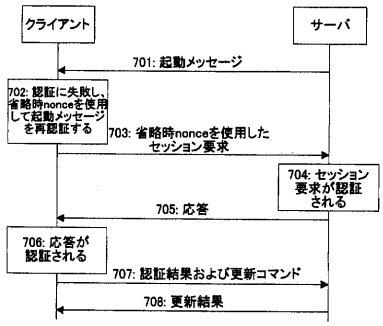
【図５】



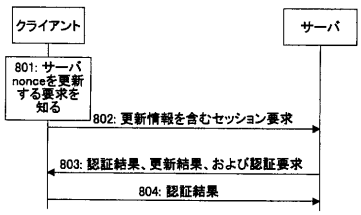
【図６】



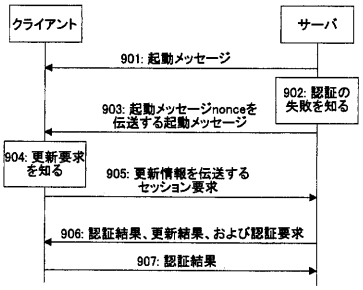
【図７】



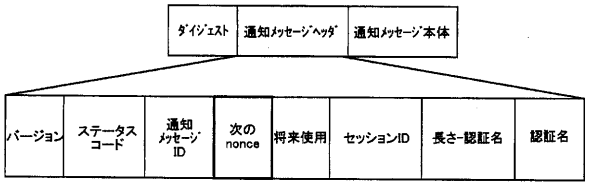
【図８】



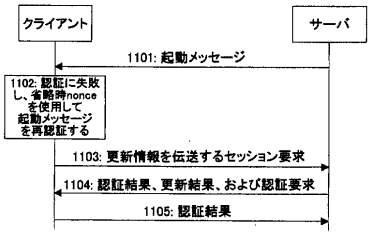
【図９】



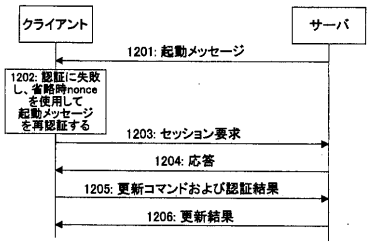
【図１０】



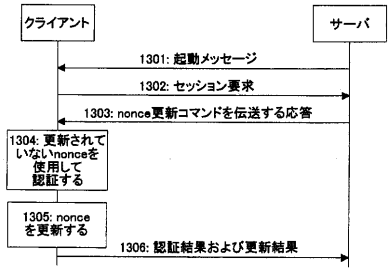
【図１１】



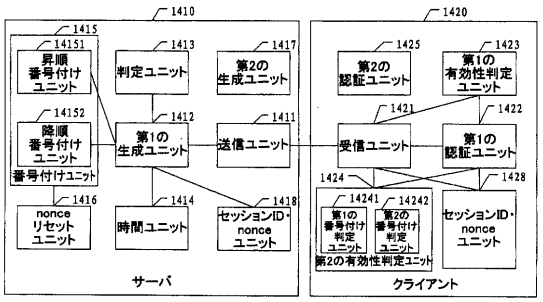
【図１２】



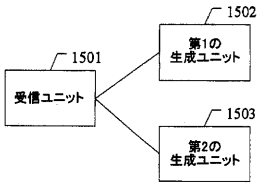
【図１３】



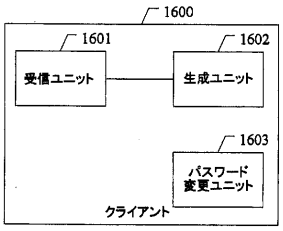
【図１４】



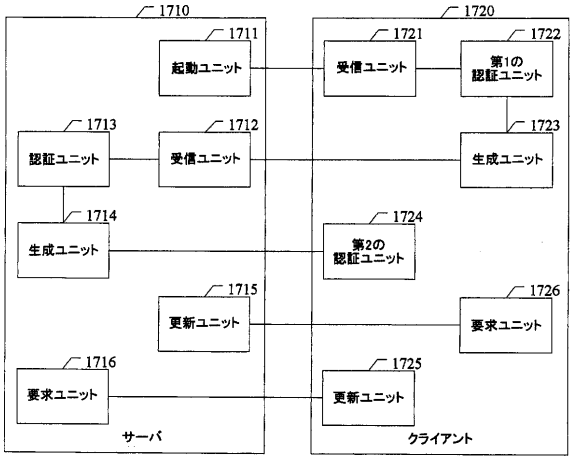
【図１５】



【図１６】



【図１７】


1710
1711
起動ユニット
1713
認証ユニット
1712
受信ユニット
1714
生成ユニット
1715
更新ユニット
1716
要求ユニット
サーバ
1720
1721
受信ユニット
1722
第1の
認証ユニット
1723
生成ユニット
1724
第2の
認証ユニット
1726
要求ユニット
1725
更新ユニット
クライアント

フロントページの続き

(72)発明者　高　洪涛
中華人民共和国５１８１２９広東省深▲セン▼市龍岡区坂田華為本社ビル
(72)発明者　李　克鵬
中華人民共和国５１８１２９広東省深▲セン▼市龍岡区坂田華為本社ビル
(72)発明者　田　林一
中華人民共和国５１８１２９広東省深▲セン▼市龍岡区坂田華為本社ビル

審査官　戸島　弘詩

(56)参考文献　特表２００４－５０９５６７（ＪＰ，Ａ）
特開２００３－３３７８６８（ＪＰ，Ａ）
特開平１１－２７５０６９（ＪＰ，Ａ）
米国特許出願公開第２００６／０１７４１０３（ＵＳ，Ａ１）
特表２００３－５１０７１３（ＪＰ，Ａ）
特開２００５－００４７６９（ＪＰ，Ａ）
米国特許第０６０６４７３６（ＵＳ，Ａ）
DRM Specification, Open Mobile Alliance, ２００６年　3月　3日, Approved Version 2.0, p.20-21,23-26, ＵＲＬ, http://member.openmobilealliance.org/ftp/Public_documents/DRM/Permanent_documents/
OMA Device Management Protocol, Open Mobile Alliance, ２００６年　6月　2日, Candidate Version 1.2, p.22-25, ＵＲＬ, http://member.openmobilealliance.org/ftp/Public_documents/DRM/Permanent_documents/
福田　洋治, ＨＴＴＰに基づくコンテンツ共有システムのアクセス制御について, コンピュータセキュリティシンポジウム２００６　論文集, 社団法人情報処理学会, ２００６年１０月２５日, 第571頁

(58)調査した分野(Int.Cl., ＤＢ名)
Ｇ０６Ｆ２１／３０，１２／００
Ｇ０９Ｃ１／００－５／００
Ｈ０４Ｋ１／００
Ｈ０４Ｌ９／００