(19)日本国特許庁(JP)　(12)公表特許公報(A)　(11)特許出願公表番号
特表2004-533751
(P2004-533751A)
(43)公表日　平成16年11月4日(2004.11.4)

(51)Int.Cl.[7]	F I		テーマコード（参考）
H04L 12/24	H04L 12/24		5B085
G06F 15/00	G06F 15/00	310D	5K030
	G06F 15/00	330A	

審査請求 未請求　予備審査請求 有　（全 151 頁）

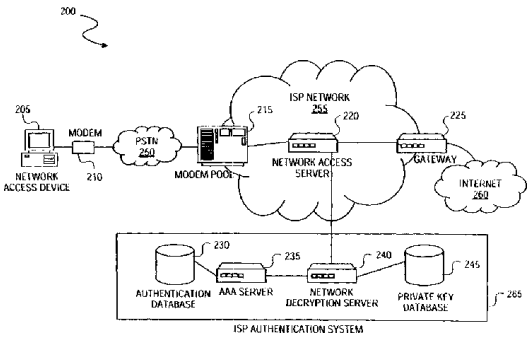
(21)出願番号	特願2002-584170 (P2002-584170)
(86)(22)出願日	平成14年4月18日 (2002.4.18)
(85)翻訳文提出日	平成15年10月20日 (2003.10.20)
(86)国際出願番号	PCT/US2002/012475
(87)国際公開番号	WO2002/086716
(87)国際公開日	平成14年10月31日 (2002.10.31)
(31)優先権主張番号	60/284,914
(32)優先日	平成13年4月18日 (2001.4.18)
(33)優先権主張国	米国 (US)
(31)優先権主張番号	10/118,380
(32)優先日	平成14年4月5日 (2002.4.5)
(33)優先権主張国	米国 (US)

(71)出願人　502309177
アイパス・インコーポレーテッド
アメリカ合衆国・９４０６５・カリフォルニア州・レッドウッド　シティ・ブリッジ　パークウェイ・３８００
(74)代理人　100064621
弁理士　山川　政樹
(72)発明者　エジェット，ジェフ・スティーブン
アメリカ合衆国・９４０８６・カリフォルニア州・サニイベイル・サウス　ベルナド・１５１・ナンバー　２４
(72)発明者　サンダー，シンガム
アメリカ合衆国・９５１３６・カリフォルニア州・サン　ノゼ・アイザック　コート・５３９

最終頁に続く

(54)【発明の名称】データ記録を関連付けるためのシステムおよび方法

(57)【要約】
複数のサービス・プロバイダ（４５２）を含めてサービス・アクセス・システムで生成される複数のトランザクション・データ記録を関連付ける方法（４００）およびシステム（２００）が提供される。トランザクション・データ記録は、単一ユーザ・セッション中にユーザがシステムにアクセスしたのに応えて生成される。この方法は、単一ユーザ・セッションに一意に関連付けられ、サービス・プロバイダによって受信可能な一意のセッションＩＤ（５２０）を生成することを含む。一意のセッションＩＤ（５２０）は、トランザクション・データ記録に含まれる。複数のトランザクション・データ記録は、トランザクション処理施設でサービス・プロバイダから受信され、各トランザクション・データ記録の一意のセッションＩＤを使用して処理する。ある実施形態では、ユーザ・セッションが認証である場合、各トランザクション・データ記録のユーザＩＤストリングに一意のセッションＩＤが提供される。



【特許請求の範囲】
【請求項１】
少なくとも１つのサービス・プロバイダを含むサービス・アクセス・システムで生成された複数のトランザクション・データ記録を関連付ける方法であって、トランザクション・データ記録は、単一ユーザ・セッション中にユーザがシステムにアクセスしたのに応えて生成され、
単一ユーザ・セッションに一意に関連付けられ、少なくとも１つのサービス・プロバイダによって受信可能であり、トランザクション・データ記録に含まれる一意のセッションＩＤを生成するステップと、
トランザクション処理ファシリティで少なくとも１つのサービス・プロバイダから複数のトランザクション・データ記録を受信するステップと、
各トランザクション・データ記録の一意のセッションＩＤを使用してトランザクション・データ記録を処理するステップと
を含む方法。
【請求項２】
ユーザ・セッションが認可される際に各トランザクション・データ記録のユーザＩＤストリングに一意のセッションＩＤを提供するステップを含む請求項１に記載の方法。
【請求項３】
単一のユーザ・セッションに一意に関連付けられる一意のコードを生成するステップと、一意のコードをユーザＩＤストリングに含めるステップとを含む請求項２に記載の方法。
【請求項４】
ユーザがアクセスを要求する接続アプリケーションで一意のコードを生成するステップと、一意のコードを、接続アプリケーションを識別する接続アプリケーションＩＤに組み合わせるステップとを含む請求項３に記載の方法。
【請求項５】
カウンターを用いて一意のコードを生成するステップを含む請求項４に記載の方法。
【請求項６】
ポイント・ツー・ポイント・プロトコル（ＰＰＰ）、パスワード認証プロトコル（ＰＡＰ）、チャレンジ・ハンドシェーク認証プロトコル（ＣＨＡＰ）、遠隔認証ダイヤルイン・ユーザ・サービス（ＲＡＤＩＵＳ）プロトコル、ターミナル・アクセス・コントローラ・アクセス制御システム（ＴＡＣＡＣＳ）プロトコル、ライトウェイト・ディレクトリ・アクセス・プロトコル（ＬＤＡＰ）、ＮＴドメイン認証プロトコル、Ｕｎｉｘパスワード認証プロトコル、ハイパーテキスト転送プロトコル（ＨＴＴＰ）、セキュア・ソケット・レイヤを介したハイパーテキスト転送プロトコル（ＨＴＴＰＳ）、拡張認証プロトコル（ＥＡＰ）、転送レイヤ・セキュリティ（ＴＬＳ）プロトコル、トークン・リング・プロトコルおよびセキュア遠隔パスワード・プロトコル（ＳＲＰ）のうちの１つのプロトコルを使用して通信に適した形式で一意のセッションＩＤを提供するステップを含む請求項３に記載の方法。
【請求項７】
最大長６３文字のユーザ・ストリング内に一意のセッションＩＤを提供するステップを含む請求項６に記載の方法。
【請求項８】
一意のコードを定義するために３桁の英数字をランダムに生成するステップと、接続アプリケーションを一意に識別する５桁の接続アプリケーションＩＤを提供するステップと、ユーザを識別する１１文字のユーザＩＤストリングを提供するステップとを含む請求項６に記載の方法。
【請求項９】
一意のセッションＩＤ、認証サービス・プロバイダからの顧客ＩＤ、顧客データ、内部顧客経路指定のための顧客経路指定データ、ユーザＩＤデータ、ユーザが内部経路指定のために使用する顧客ドメイン・データ、およびトランザクション・データ記録の非経路指定

データ顧客データの少なくとも１つから変更されたトランザクション記録データを構築するステップを含む請求項６に記載の方法。
【請求項１０】
ユーザがシステムによって肯定的に認証された場合にのみセッションの開始を許可するステップと、肯定的な認証の際に、トランザクション処理ファシリティの認証トランザクション記憶領域内のセッションＩＤフィールドに一意のセッションＩＤを記憶するステップとを含む請求項３に記載の方法。
【請求項１１】
少なくとも１つのサービス・プロバイダからトランザクション・データ記録を取り出すステップと、各トランザクション・データ記録を一意のセッションＩＤデータに基づいてアカウンティング・トランザクション記憶領域に記憶するステップとを含む請求項１０に記載の方法。
【請求項１２】
少なくとも１つのサービス・プロバイダからバッチ・ローディング・トランザクション・データ記録を定期的に受信するステップと、変更されたセッションＩＤデータ記録を構築するステップと、変更されたセッションＩＤデータ記録をバッチ履歴記録領域内のセッションＩＤフィールドに記憶するステップとを含む請求項１１に記載の方法。
【請求項１３】
サービス品質モニタ（ＳＱＭ）トランザクション・データ記録を受信するステップと、ＳＱＭデータ記録から変更されたトランザクション・データ記録を構築するステップと、変更されたトランザクション・データ記録をＳＱＭ記憶領域内のセッションＩＤフィールドに記憶するステップとを含む請求項１２に記載の方法。
【請求項１４】
欠落したアカウンティング記録を識別するために認証トランザクション記憶領域とアカウンティング・トランザクション領域内のセッションＩＤデータを比較するステップを含む請求項１１に記載の方法。
【請求項１５】
認証されなかった少なくとも１つのサービス・プロバイダによって提供されたトランザクション・データ記録を識別するために、アカウンティング・トランザクション記憶領域内の一意のセッションＩＤなしに各トランザクション・データ記録を識別するステップを含む請求項１１に記載の方法。
【請求項１６】
重複したトランザクション・データ記録を識別するためにアカウンティング・トランザクション記憶領域で重複したセッションＩＤデータを検索するステップを含む請求項１１に記載の方法。
【請求項１７】
重複したエイリアス記録、ＩＳＤＮデュアル・チャネル記録、無効なセッション長記録、およびオーバーラップしているアカウンティング記録の少なくとも１つを識別するために一意のセッションＩＤを使用するステップを含む請求項１１に記載の方法。
【請求項１８】
少なくとも１つのサービス・プロバイダを含むサービス・アクセス・システムで生成されたトランザクション・データ記録を処理するシステムであって、トランザクション・データ記録は、単一ユーザ・セッション中にユーザがシステムにアクセスしたのに応えて生成され、
単一ユーザ・セッションに一意に関連付けられ、少なくとも１つのサービス・プロバイダによって受信可能であり、トランザクション・データ記録に含まれる一意のセッションＩＤを生成するセッションＩＤジェネレータと、
各記録の一意のセッションＩＤを使用して少なくとも１つのサービス・プロバイダから受信した複数のトランザクション・データ記録を処理するトランザクション処理ファシリティと

を含むシステム。
【請求項１９】
セッションＩＤジェネレータが、ユーザ・セッションが認可される際に各トランザクション・データ記録のユーザＩＤストリングに一意のセッションＩＤを提供する請求項１８に記載のシステム。
【請求項２０】
セッションＩＤジェネレータが、単一のユーザ・セッションに一意に関連付けられる一意のコードを生成し、一意のコードがユーザＩＤストリングに含まれる請求項１９に記載のシステム。
【請求項２１】
セッションＩＤジェネレータはユーザがアクセスを要求する接続アプリケーションでソフトウェア・アプリケーションによって実施され、一意のコードが接続アプリケーションを識別する接続アプリケーションＩＤに組み合わされる請求項２０に記載のシステム。
【請求項２２】
セッションＩＤジェネレータは一意のコードを生成するカウンターである請求項２１に記載のシステム。
【請求項２３】
ポイント・ツー・ポイント・プロトコル（ＰＰＰ）、パスワード認証プロトコル（ＰＡＰ）、チャレンジ・ハンドシェーク認証プロトコル（ＣＨＡＰ）、遠隔認証ダイヤルイン・ユーザ・サービス（ＲＡＤＩＵＳ）プロトコル、ターミナル・アクセス・コントローラ・アクセス制御システム（ＴＡＣＡＣＳ）プロトコル、ライトウェイト・ディレクトリ・アクセス・プロトコル（ＬＤＡＰ）、ＮＴドメイン認証プロトコル、Ｕｎｉｘパスワード認証プロトコル、ハイパーテキスト転送プロトコル（ＨＴＴＰ）、セキュア・ソケット・レイヤを介したハイパーテキスト転送プロトコル（ＨＴＴＰＳ）、拡張認証プロトコル（ＥＡＰ）、転送レイヤ・セキュリティ（ＴＬＳ）プロトコル、トークン・リング・プロトコルおよびセキュア遠隔パスワード・プロトコル（ＳＲＰ）のうちの１つのプロトコルを含み、一意のセッションＩＤは通信で使用するために適した形式で提供される請求項２０に記載のシステム。
【請求項２４】
最大長６３文字のユーザ・ストリング内に一意のセッションＩＤを提供する請求項２３に記載のシステム。
【請求項２５】
一意のコードはランダムに生成された３桁の英数字によって提供され、接続アプリケーションＩＤは接続アプリケーションを一意に識別する５桁によって提供され、ユーザを識別するユーザＩＤストリングが１１文字によって提供される請求項２４に記載のシステム。
【請求項２６】
ユーザがシステムによって肯定的に認証された場合にのみセッションの開始を許可し、肯定的な認証の際に、トランザクション処理ファシリティは認証トランザクション記憶領域内のセッションＩＤフィールドに一意のセッションＩＤを記憶する請求項２０に記載のシステム。
【請求項２７】
トランザクション処理ファシリティが少なくとも１つのサービス・プロバイダからトランザクション・データ記録を受信し、各トランザクション・データ記録を一意のセッションＩＤデータに基づいてアカウンティング・トランザクション記憶領域に記憶する請求項２６に記載のシステム。
【請求項２８】
トランザクション処理ファシリティが、少なくとも１つのサービス・プロバイダからバッチ・ローディング・トランザクション・データ記録を定期的に受信し、変更されたセッションＩＤデータ記録を構築し、変更されたセッションＩＤデータ記録をバッチ履歴記録領域内のセッションＩＤフィールドに記憶する請求項２７に記載のシステム。

【請求項２９】
トランザクション処理ファシリティがサービス品質モニタ（ＳＱＭ）トランザクション・データ記録を受信し、ＳＱＭトランザクション・データ記録から変更されたトランザクション・データ記録を構築し、変更されたトランザクション・データ記録をＳＱＭ記憶領域内のセッションＩＤフィールドに記憶する請求項２８に記載のシステム。
【請求項３０】
少なくとも１つのサービス・プロバイダを含むサービス・アクセス・システムで生成された複数のトランザクション・データ記録を処理する方法であって、
少なくとも１つのサービス・プロバイダからトランザクション・データ記録を受信するステップであって、各トランザクション・データ記録は、単一ユーザ・セッション中にユーザがシステムにアクセスしたのに応えて生成されるステップと、
単一ユーザ・セッションに関連付けられたトランザクション・データ記録を各トランザクション・データ記録に含まれるセッションＩＤに基づいて識別するステップであって、各セッションＩＤは単一ユーザ・セッションを一意に識別するステップと
を含む方法。
【請求項３１】
各トランザクション記録のユーザＩＤストリングの一意のセッションＩＤを識別するステップを含む請求項３０に記載の方法。
【請求項３２】
単一のユーザ・セッションに一意に関連付けられ、ユーザＩＤストリングに含まれる一意のコードを識別するステップを含む請求項３１に記載の方法。
【請求項３３】
ユーザがアクセスを要求する接続アプリケーションを識別し、一意のコードを生成した接続アプリケーションＩＤを識別するステップを含む請求項３２に記載の方法。
【請求項３４】
ポイント・ツー・ポイント・プロトコル（ＰＰＰ）、パスワード認証プロトコル（ＰＡＰ）、チャレンジ・ハンドシェーク認証プロトコル（ＣＨＡＰ）、遠隔認証ダイヤルイン・ユーザ・サービス（ＲＡＤＩＵＳ）プロトコル、ターミナル・アクセス・コントローラ・アクセス制御システム（ＴＡＣＡＣＳ）プロトコル、ライトウェイト・ディレクトリ・アクセス・プロトコル（ＬＤＡＰ）、ＮＴドメイン認証プロトコル、Ｕｎｉｘパスワード認証プロトコル、ハイパーテキスト転送プロトコル（ＨＴＴＰ）、セキュア・ソケット・レイヤを介したハイパーテキスト転送プロトコル（ＨＴＴＰＳ）、拡張認証プロトコル（ＥＡＰ）、転送レイヤ・セキュリティ（ＴＬＳ）プロトコル、トークン・リング・プロトコルおよびセキュア遠隔パスワード・プロトコル（ＳＲＰ）のうちの１つのプロトコルを使用する一意のセッションＩＤを含むトランザクション・データ記録を受信する請求項３２に記載の方法。
【請求項３５】
最大長６３文字のユーザ・ストリングから一意のセッションＩＤを抽出する請求項３４に記載の方法。
【請求項３６】
３桁の英数字から一意のコードを識別し、５桁の接続アプリケーションＩＤから接続アプリケーションを識別し、１１文字のユーザＩＤストリングからユーザを識別する請求項３５に記載の方法。
【請求項３７】
一意のセッションＩＤ、認証サービス・プロバイダからの顧客ＩＤ、顧客データ、内部顧客経路指定のための顧客経路指定データ、ユーザＩＤデータ、ユーザが内部経路指定のために使用する顧客ドメイン・データ、およびトランザクション・データ記録の非経路指定データ顧客データの少なくとも１つから変更されたトランザクション記録データを構築する請求項３４に記載の方法。
【請求項３８】

トランザクション処理ファシリティの認証トランザクション記憶領域内のセッションＩＤフィールドに一意のセッションＩＤを記憶する請求項３２に記載の方法。
【請求項３９】
サービス・プロバイダから受信した各トランザクション・データ記録を一意のセッションＩＤデータに基づいてアカウンティング・トランザクション記憶領域に記憶する請求項３２に記載の方法。
【請求項４０】
少なくとも１つのサービス・プロバイダからバッチ・ローディング・トランザクション・データ記録を定期的に受信するステップと、トランザクション・データ記録から変更されたセッションＩＤデータ記録を構築するステップと、変更されたセッションＩＤデータ記録をバッチ履歴記録領域内のセッションＩＤフィールドに記憶するステップを含む請求項３９に記載の方法。
【請求項４１】
サービス品質モニタ（ＳＱＭ）トランザクション・データ記録を受信するステップと、変更されたトランザクション・データ記録から変更されたトランザクション・データ記録を構築するステップと、変更されたトランザクション・データ記録をＳＱＭ記憶領域内のセッションＩＤフィールドに記憶するステップとを含む請求項３９に記載の方法。
【請求項４２】
欠落したアカウンティング記録を識別するために認証トランザクション記憶領域とアカウンティング・トランザクション領域内のセッションＩＤデータを比較するステップを含む請求項３９に記載の方法。
【請求項４３】
認証されなかった少なくとも１つのサービス・プロバイダによって提供されたトランザクション・データ記録を識別するために、アカウンティング・トランザクション記憶領域内の一意のセッションＩＤなしに各トランザクション・データ記録を識別するステップを含む請求項３９に記載の方法。
【請求項４４】
重複したトランザクション・データ記録を識別するためにアカウンティング・トランザクション記憶領域で重複したセッションＩＤデータを検索するステップを含む請求項３９に記載の方法。
【請求項４５】
重複したエイリアス記録、ＩＳＤＮデュアル・チャネル記録、無効なセッション長記録、およびオーバーラップしているアカウンティング記録の少なくとも１つを識別するために一意のセッションＩＤを使用するステップを含む請求項３９に記載の方法。
【請求項４６】
少なくとも１つのサービス・プロバイダを含むサービス・アクセス・システムで生成された複数のトランザクション・データ記録を処理するトランザクション処理ファシリティであって、
少なくとも１つのサービス・プロバイダからトランザクション・データ記録を受信し、各トランザクション・データ記録は、単一ユーザ・セッション中にユーザがシステムにアクセスしたのに応えて生成され、
単一ユーザ・セッションに関連付けられたトランザクション・データ記録を各トランザクション・データ記録に含まれるセッションＩＤに基づいて識別し、各セッションＩＤは単一ユーザ・セッションを一意に識別する
ように構成されたトランザクション処理ファシリティ。
【請求項４７】
各トランザクション・データ記録のユーザＩＤストリングで一意のセッションＩＤを識別する請求項４６に記載のトランザクション処理ファシリティ。
【請求項４８】
単一のユーザ・セッションに一意に関連付けられ、ユーザＩＤストリングに含まれる一意

のコードを識別する請求項４７に記載のトランザクション処理ファシリティ。
【請求項４９】
一意のコードを生成した接続アプリケーションを識別し、ユーザがそれによってアクセスを得る接続アプリケーションＩＤを識別する請求項４７に記載のトランザクション処理ファシリティ。
【請求項５０】
ポイント・ツー・ポイント・プロトコル（ＰＰＰ）、パスワード認証プロトコル（ＰＡＰ）、チャレンジ・ハンドシェーク認証プロトコル（ＣＨＡＰ）、遠隔認証ダイヤルイン・ユーザ・サービス（ＲＡＤＩＵＳ）プロトコル、ターミナル・アクセス・コントローラ・アクセス制御システム（ＴＡＣＡＣＳ）プロトコル、ライトウェイト・ディレクトリ・アクセス・プロトコル（ＬＤＡＰ）、ＮＴドメイン認証プロトコル、Ｕｎｉｘパスワード認証プロトコル、ハイパーテキスト転送プロトコル（ＨＴＴＰ）、セキュア・ソケット・レイヤを介したハイパーテキスト転送プロトコル（ＨＴＴＰＳ）、拡張認証プロトコル（ＥＡＰ）、転送レイヤ・セキュリティ（ＴＬＳ）プロトコル、トークン・リング・プロトコルおよびセキュア遠隔パスワード・プロトコル（ＳＲＰ）のうちの１つのプロトコルを使用して一意のセッションＩＤを含めてトランザクション・データ記録を受信する請求項４８に記載のトランザクション処理ファシリティ。
【請求項５１】
最大長６３文字のユーザ・ストリングから一意のセッションＩＤを抽出する請求項５０に記載のトランザクション処理ファシリティ。
【請求項５２】
３桁の英数字から一意のコードを識別し、５桁の接続アプリケーションＩＤから接続アプリケーションを識別し、１１文字のユーザＩＤストリングからユーザを識別する請求項５１に記載のトランザクション処理ファシリティ。
【請求項５３】
一意のセッションＩＤ、認証サービス・プロバイダからの顧客ＩＤ、顧客データ、内部顧客経路指定のための顧客経路指定データ、ユーザＩＤデータ、ユーザが内部経路指定のために使用する顧客ドメイン・データ、およびトランザクション・データ記録の非経路指定データ顧客データの少なくとも１つから変更されたトランザクション記録データを構築する請求項４７に記載のトランザクション処理ファシリティ。
【請求項５４】
トランザクション処理ファシリティの認証トランザクション記憶領域内のセッションＩＤフィールドに一意のセッションＩＤを記憶する請求項４７に記載のトランザクション処理ファシリティ。
【請求項５５】
少なくとも１つのサービス・プロバイダから受信した各トランザクション・データ記録を一意のセッションＩＤデータに基づいてアカウンティング・トランザクション記憶領域に記憶する請求項４７に記載のトランザクション処理ファシリティ。
【請求項５６】
少なくとも１つのサービス・プロバイダからバッチ・ローディング・トランザクション・データ記録を定期的に受信し、そこからバッチ履歴記録領域内のセッションＩＤフィールドに記憶されている変更されたセッションＩＤデータ記録を構築する請求項５５に記載のトランザクション処理ファシリティ。
【請求項５７】
サービス品質モニタ（ＳＱＭ）トランザクション・データ記録を受信し、ＳＱＭトランザクション・データ記録から変更されたトランザクション・データ記録を構築し、ＳＱＭ記憶領域内のセッションＩＤフィールドにそれらを記憶する請求項５５に記載のトランザクション処理ファシリティ。
【請求項５８】
欠落したアカウンティング記録を識別するために認証トランザクション記憶領域とアカウ

ンティング・トランザクション領域内のセッションＩＤデータを比較する請求項５５に記載のトランザクション処理ファシリティ。
【請求項５９】
認証されなかった少なくとも１つのサービス・プロバイダによって提供されたトランザクション・データ記録を識別するために、アカウンティング・トランザクション記憶領域内の一意のセッションＩＤなしに各トランザクション・データ記録を識別する請求項５５に記載のトランザクション処理ファシリティ。
【請求項６０】
重複したトランザクション・データ記録を識別するためにアカウンティング・トランザクション記憶領域で重複したセッションＩＤデータを検索する請求項５５に記載のトランザクション処理ファシリティ。
【請求項６１】
重複したエイリアス記録、ＩＳＤＮデュアル・チャネル記録、無効なセッション長記録、およびオーバーラップしているアカウンティング記録の少なくとも１つを識別するために一意のセッションＩＤを使用する請求項５５に記載のトランザクション処理ファシリティ。
【請求項６２】
ユーザをアクセス・サービス・プロバイダに接続する方法であって、ユーザがサービス・プロバイダにアクセスする単一ユーザ・セッションに関連付けられた一意のセッションＩＤを作成するステップであって、ユーザ・セッションが認可される際に一意のセッションＩＤが各トランザクション・データ記録のユーザＩＤストリングに提供されるステップを含む方法。
【請求項６３】
単一のユーザ・セッションに一意に関連付けられる一意のコードを生成するステップと、一意のコードをユーザＩＤストリングに含めるステップとを含む請求項６２に記載の方法。
【請求項６４】
ユーザがアクセスを得る接続アプリケーションで一意のコードを生成するステップと、一意のコードを、接続アプリケーションを識別する接続アプリケーションＩＤと組み合わせるステップとを含む請求項６３に記載の方法。
【請求項６５】
カウンターを用いて一意のコードを生成するステップを含む請求項６４に記載の方法。
【請求項６６】
ポイント・ツー・ポイント・プロトコル（ＰＰＰ）、パスワード認証プロトコル（ＰＡＰ）、チャレンジ・ハンドシェーク認証プロトコル（ＣＨＡＰ）、遠隔認証ダイヤルイン・ユーザ・サービス（ＲＡＤＩＵＳ）プロトコル、ターミナル・アクセス・コントローラ・アクセス制御システム（ＴＡＣＡＣＳ）プロトコル、ライトウェイト・ディレクトリ・アクセス・プロトコル（ＬＤＡＰ）、ＮＴドメイン認証プロトコル、Ｕｎｉｘパスワード認証プロトコル、ハイパーテキスト転送プロトコル（ＨＴＴＰ）、セキュア・ソケット・レイヤを介したハイパーテキスト転送プロトコル（ＨＴＴＰＳ）、拡張認証プロトコル（ＥＡＰ）、転送レイヤ・セキュリティ（ＴＬＳ）プロトコル、トークン・リング・プロトコルおよびセキュア遠隔パスワード・プロトコル（ＳＲＰ）のうちの１つのプロトコルを使用して通信に適した形式で一意のセッションＩＤを提供するステップを含む請求項６３に記載の方法。
【請求項６７】
最大長６３文字のユーザ・ストリング内に一意のセッションＩＤを提供するステップを含む請求項６６に記載の方法。
【請求項６８】
一意のコードを定義するために３桁の英数字をランダムに生成するステップと、接続アプリケーションを一意に識別する５桁の接続アプリケーションＩＤを提供するステップと、

ユーザを識別する１１文字のユーザＩＤストリングを提供するステップとを含む請求項６７に記載の方法。
【請求項６９】
ユーザをアクセス・サービス・プロバイダに接続する接続装置であって、認可された各セッションに関連付けられた一意のセッションＩＤを作成するセッションＩＤジェネレータを含む接続装置。
【請求項７０】
セッションＩＤジェネレータが単一のユーザ・セッションに一意に関連付けられる一意のコードを生成し、一意のコードがユーザＩＤストリングに含まれる請求項６９に記載の接続装置。
【請求項７１】
セッションＩＤジェネレータが、一意のコードを、接続アプリケーションを識別する接続装置ＩＤと組み合わせる請求項７０に記載の接続装置。
【請求項７２】
セッションＩＤジェネレータが、カウンターを用いて一意のコードを生成する請求項７０に記載の接続装置。
【請求項７３】
ポイント・ツー・ポイント・プロトコル（ＰＰＰ）、パスワード認証プロトコル（ＰＡＰ）、チャレンジ・ハンドシェーク認証プロトコル（ＣＨＡＰ）、遠隔認証ダイヤルイン・ユーザ・サービス（ＲＡＤＩＵＳ）プロトコル、ターミナル・アクセス・コントローラ・アクセス制御システム（ＴＡＣＡＣＳ）プロトコル、ライトウェイト・ディレクトリ・アクセス・プロトコル（ＬＤＡＰ）、ＮＴドメイン認証プロトコル、Ｕｎｉｘパスワード認証プロトコル、ハイパーテキスト転送プロトコル（ＨＴＴＰ）、セキュア・ソケット・レイヤを介したハイパーテキスト転送プロトコル（ＨＴＴＰＳ）、拡張認証プロトコル（ＥＡＰ）、転送レイヤ・セキュリティ（ＴＬＳ）プロトコル、トークン・リング・プロトコルおよびセキュア遠隔パスワード・プロトコル（ＳＲＰ）のうちの１つのプロトコルを使用して通信に適した形式で一意のセッションＩＤが提供される請求項７０に記載の接続装置。
【請求項７４】
最大長６３文字のユーザ・ストリング内に一意のセッションＩＤが提供される請求項７３に記載の接続装置。
【請求項７５】
セッションＩＤジェネレータが、一意のコードを定義するために３桁の英数字をランダムに生成し、接続装置を一意に識別する５桁の接続アプリケーションＩＤを提供し、ユーザを識別する１１文字のユーザＩＤストリングを提供する請求項７４に記載の接続装置。
【請求項７６】
少なくとも１つのサービス・プロバイダを含むサービス・アクセス・システムで生成された複数のトランザクション・データ記録を関連付ける一連の命令を実装する機械可読媒体であって、トランザクション・データ記録は、単一ユーザ・セッション中にユーザがシステムにアクセスしたのに応えて生成され、命令は、機械によって実行されると、機械に
単一ユーザ・セッションに一意に関連付けられ、少なくとも１つのサービス・プロバイダによって受信可能であり、トランザクション・データ記録に含まれる一意のセッションＩＤを生成させ、
トランザクション処理ファシリティでサービス・プロバイダから複数のトランザクション・データ記録を受信させ、
各トランザクション・データ記録の一意のセッションＩＤを使用してトランザクション・データ記録を処理させる
機械可読媒体。
【請求項７７】
ユーザ・セッションが認可される際に各トランザクション・データ記録のユーザＩＤスト

リングに一意のセッションＩＤが提供される請求項７６に記載の機械可読媒体。
【請求項７８】
単一のユーザ・セッションに一意に関連付けられる一意のコードが生成され、ユーザＩＤストリングに含められる請求項７７に記載の機械可読媒体。
【請求項７９】
少なくとも１つのサービス・プロバイダを含むサービス・アクセス・システムで生成された複数のトランザクション・データ記録を処理する一連の命令を実装した機械可読媒体であって、命令は、機械によって実行されると、機械に
少なくとも１つのプロバイダからトランザクション・データ記録を受信させ、各トランザクション・データ記録は、単一ユーザ・セッション中にユーザがシステムにアクセスしたのに応えて生成され、
単一ユーザ・セッションに関連付けられたトランザクション・データ記録を各トランザクション・データ記録に含まれるセッションＩＤに基づいて識別させ、各セッションＩＤは単一ユーザ・セッションを一意に識別する
ことを行わせる機械可読媒体。
【請求項８０】
ユーザをアクセス・プロバイダに接続することに関する一連の命令を実装する機械可読媒体であって、命令は、機械によって実行されると、機械に、ユーザがサービス・プロバイダにアクセスする単一ユーザ・セッションに関連付けられた一意のセッションＩＤを作成させ、一意のセッションＩＤが、ユーザ・セッションが認可される際に各トランザクション・データ記録のユーザＩＤストリングに提供されることを行わせる機械可読媒体。
【請求項８１】
少なくとも１つのサービス・プロバイダを含むサービス・アクセス・システムで生成された複数のトランザクション・データ記録を処理するトランザクション処理ファシリティであって、
少なくとも１つのサービス・プロバイダからトランザクション・データ記録を受信する受信手段であって、各トランザクション・データ記録が、単一ユーザ・セッション中にユーザがシステムにアクセスしたのに応えて生成される受信手段と、
単一ユーザ・セッションに関連付けられたトランザクション・データ記録を各トランザクション・データ記録に含まれるセッションＩＤに基づいて識別するプロセッサ手段であって、各セッションＩＤが単一ユーザ・セッションを一意に識別するプロセッサ手段と
を含むトランザクション処理ファシリティ。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、一般に、ネットワーク・アクセス環境でトランザクション記録を処理することに関する。より詳細には、本発明は、サービス・アクセス・システムで生成される複数のトランザクション・データ記録を関連付ける方法およびシステムに関する。
【背景技術】
【０００２】
経済活動の国際化が進んでいるため、地理的に分散した人および施設の間に通信を提供する必要性が高まっている。例えば、特定の企業は、複数の国家および大陸にわたって施設を設置していることがある。国際化は、さらに出張の増加をもたらした。インターネット・ベースの通信に企業や個人がますます依存するようになることによって、移動の多い社員（いわゆる「出張族」）が海外出張時にインターネット・ベースの無線通信にアクセスできることがさらに望まれるようになってきた。そのような移動の多い人物に対して通信を容易にするサービスは、一般に「ローミング・サービス」と呼ばれている。インターネット・ベースの通信を一例と見なすと、モバイル顧客の要求に応えるために、インターネット・サービス・プロバイダ（ＩＳＰ）は、「ローミング」インターネット・アクセス・ソリューションと呼ばれているサービスのような、世界中の様々な場所からインターネッ

トにローカル・コール・アクセスを提供することを開始している。ローミング・ソリューションが求められる主たる理由は、ＩＳＰが地理的領域ごとに特化される傾向があり、これがサービスの範囲に差異を生じさせていることである。要求される信頼性および性能基準を満たすネットワーク・インフラストラクチャ、ネットワーク管理、および継続的なアップグレードの拡張はどれも、ＩＳＰに対して資金と時間の両面で多大な重荷を科すものである。これらの理由から、多くのＩＳＰは、限られた地理的領域内にポインツ・オブ・プレゼンス（ＰＯＰ）を置くだけだった。

【０００３】

上記の理由により、多くの企業がその遠距離通勤者と移動の多い社員に対して従来の遠隔アクセス・ソリューションに取って代わるようにインターネット・ベースの通信を利用するので、ＩＳＰがインターネット・ローミング・ソリューションを、特に法人顧客に対して提供する機能の重要性はますます高まっている。

【０００４】

インターネット・ローミング・ソリューションを提供するために、一部のＩＳＰはさらに地理的な到達範囲を広げるためにネットワーク・インフラストラクチャを共有することを開始した。このようなインフラストラクチャの共有は、１つのＩＳＰのユーザが別のＩＳＰのネットワークを介したインターネット・アクセスを行うことができる協定の形式を取る場合がある。世界的な到達範囲を提供する試みにおいてＩＳＰ同士が提携する数が増加するに伴い、アカウンティング情報を管理し処理し、コストを共有することが複雑化することが理解されよう。例えば、比較的少数のＩＳＰであっても、欠落したアカウンティング記録、不適切なアカウンティング記録、重複したアカウンティング記録、重複したエイリアス記録、無効なセッション長記録、オーバーラップしたアカウンティング記録などに関してしばしば問題が発生する。

【０００５】

通常、ユーザはインターネット・サービス・プロバイダ（ＩＳＰ）を介してインターネットにアクセスする。インターネットまたは企業のローカル・エリア・ネットワーク（ＬＡＮ）にアクセスしようとしているネットワーク・ユーザは、一般にＩＤを検証するためにユーザ名とパスワードを入力する必要がある。この過程に関する問題は、多くの標準認証プロトコルを使用しているＩＳＰに送信する際にパスワードが一般的に安全ではないということである。

【０００６】

図１は従来技術のＩＳＰネットワーク構成１００を示す図であるが、ここではネットワーク・ユーザの信用証明書が安全でない方法で認証されている。ＩＳＰネットワーク１４５は、モデム・プール１１５に接続されており、またゲートウェイ１２５を介してインターネット１５０に接続されているネットワーク・アクセス・サーバ（ＮＡＳ）１２０を含む。ＩＳＰネットワーク１４５は、認証サーバ（ＡＡＡサーバ）１３５にも接続されている。ＡＡＡサーバ１３５は、ＩＳＰネットワーク１４５に局所的にあってもよく、あるいはＩＳＰネットワーク１４５から非常に離れた遠隔位置にあってもよい。

【０００７】

インターネット接続を確立するために、ネットワーク・ユーザは、通常、ネットワーク・アクセス・デバイス１０５でダイヤルアップ・ネットワーキング・アプリケーションを実行する。ダイヤルアップ・ネットワーキング・アプリケーションは、公衆交換電話網（ＰＳＴＮ）１４０を介してモデム・プール１１５とモデム・セッションを開始するために、ネットワーク・ユーザ名とネットワーク・パスワードを入力し、モデム１１０を操作するようユーザを促す。モデム・セッションが確立されると、ダイヤルアップ・ネットワーキング・アプリケーションは、データ接続を確立し、ネットワーク・ユーザを認証するためにＮＡＳ１２０との通信を開始する。

【０００８】

コンピュータ間で接続を確立するために使用されるさらに一般的なデータ通信プロトコルの１つはポイント・ツー・ポイント・プロトコル（ＰＰＰ）である。一般にＰＰＰと共に

使用される１つの特に良く知られた認証プロトコルは、パスワード認証プロトコル（ＰＡＰ）である。ＰＡＰを使用するよう構成されたダイヤルアップ・ネットワーキング・アプリケーションは、認可確認応答信号が受信されるか、または接続が終了するまで、確立したデータ接続を介してユーザ名とパスワードの対を繰り返し送信する。ダイヤルアップ・ネットワーキング・アプリケーションは、ユーザ名とパスワードを送信する頻度とタイミングを制御するよう構成されている。

【０００９】

ＰＡＰに関する問題は、パスワードがデータ接続を介して送信される前に暗号化されず、平文として送信されるということである。つまり、パスワードがハッカーによる傍受を受けやすいということである。例えば、データ接続にアクセスするハッカーは、ネットワーク監視アプリケーションを使用して、データ接続全体で送信されているデータ・パケットを補足し表示することができる。このようなネットワーク監視アプリケーションは一般的であり、その使用が違法であるためにこれらはしばしばパケット・スニッフィング・アプリケーションまたはパケット・スヌーピング・アプリケーションと呼ばれている。

【００１０】

再度図１を参照すると、ＮＡＳ１２０でユーザ名とパスワードの対が一度受信されると、通常、別の標準認証プロトコルであるリモート認証ダイヤルイン・ユーザ・サービス（ＲＡＤＩＵＳ）が、ＩＳＰ認証システム１５５にネットワーク・ユーザ名とパスワードの対を送信するために使用される。ＲＡＤＩＵＳプロトコルは、パスワードがＩＳＰ認証システム１５５のＡＡＡサーバ１３５に送信される前にそのパスワードの対称暗号化を提供する。この暗号化方法は、ＮＡＳ１２０とＡＡＡサーバ１３５が暗号化アルゴリズムで使用される秘密鍵を共有するので対称と見なされる。ＮＡＳ１２０は、パスワードを「ロック」すなわち暗号化するために秘密鍵を使用し、ＡＡＡサーバ１３５は、当該パスワードを認証データベース１３０に記憶されているパスワードと照合する前に当該パスワードを「アンロック」すなわち暗号解読するために秘密鍵を使用する。

【００１１】

ＲＡＤＩＵＳ対称暗号化方法に関する問題は、「辞書」攻撃として知られている一形態の攻撃を受けやすいということである。辞書攻撃では、暗号化アルゴリズムの知識のあるハッカーは、暗号化されたパスワードをパケット・スニッフィング・アプリケーションによって傍受する。次いでハッカーは、可読文字を生じる鍵が見つかるまで一連の鍵を繰り返し試みる。さらに問題を深刻化させるのは、秘密鍵が一度漏洩すると、ＮＡＳ１２０とＡＡＡサーバ１３５の間で傍受したいかなるパスワードでもハッカーは容易に暗号解読することができるということである。

【００１２】

上述のＰＡＰ／ＲＡＤＩＵＳ認証方法に固有の弱点を受けて、チャレンジ・ハンドシェーク認証プロトコル（ＣＨＡＰ）が開発された。ＣＨＡＰを使用するように実装されたシステムで、ネットワーク・アクセス・デバイス１０５のダイヤルアップ・アプリケーションは、認証プロトコルとしてＰＡＰではなくＣＨＡＰを使用することをＮＡＳと交渉する。次にＮＡＳ１２０は乱数を生成し、それをネットワーク・アクセス・デバイス１０５に送信する。ネットワーク・アクセス・デバイス１０５で実行されているダイヤルアップ・ネットワーキング・アプリケーションは、乱数を使用してパスワードのノンリバーシブル・ハッシュを生成し、そのノンリバーシブル・ハッシュは次いでＮＡＳ１２０に送信される。次いでＮＡＳ１２０は、ＲＡＤＩＵＳプロトコルを使用して、ノンリバーシブル・ハッシュとそのハッシュを生成するために使用された乱数をＡＡＡサーバ１３５に送信する。ＡＡＡサーバ１３５は、認証データベース１３０から平文パスワードを取り出し、ＮＡＳ１２０から受信した乱数を使用してハッシュ・オペレーションを繰り返す。最後に、ＡＡＡサーバ１３５は、その生成されたハッシュ値をＮＡＳ１２０から受信したハッシュ値と比較する。ハッシュ値が同一である場合、認証は成功と見なされ、ＡＡＡサーバ１３５はネットワーク・アクセス・デバイス１０５に適切な確認応答信号を送信する。

【発明の開示】

【発明が解決しようとする課題】
【００１３】
ユーザ認証のためのＣＨＡＰ／ＲＡＤＩＵＳ方法に関する問題は、これら３つのシステム、すなわちネットワーク・アクセス・デバイス１０５、ＮＡＳ１２０、およびＡＡＡサーバ１３５は、安全性が付加されたことを利用するためにＣＨＡＰを使用するよう構成しなければならないということである。これら３つのうちどれかがＣＨＡＰを使用するように構成されていない場合、ネットワーク・アクセス・デバイス１０５のダイヤルアップ・ネットワーキング・アプリケーションは、ＰＡＰを認証プロトコルとして使用することをＮＡＳ１２０と交渉するためにＰＰＰプロトコルを使用する。
【００１４】
ＣＨＡＰ／ＲＡＤＩＵＳ方法を使用することのもう１つの不利な点は、ＣＨＡＰを適切に実施するために、ＡＡＡサーバ１３５は平文パスワードにアクセスしなければならないということである。多くの認証システムは、システムに漏洩が発生しパスワードが盗まれた場合に安全性のリスクが増すので、パスワードは平文形式では記憶しない。
【００１５】
さらに最近では、認証システムは、拡張認証プロトコル（ＥＡＰ）と呼ばれる認証プロトコルを採用している。ＥＡＰは、パスワードをハッシュするためにネットワーク・アクセス・デバイス１０５が使用する乱数をＮＡＳ１２０ではなくＡＡＡサーバ１３５が生成するということを除いて、ＣＨＡＰと略同様に機能する。その結果、ＥＡＰはＣＨＡＰと同じ短所を有することになる。具体的には、認証チェーンのすべてのシステムがＥＡＰを採用している場合にだけＥＡＰは有効である。
【００１６】
ブロードバンド・アクセスの出現により、無線のアクセス・プロバイダと有線（イーサネット（登録商標））のアクセス・プロバイダはどちらも、ウェブ・ブラウザ・ベースの認証システムを採用する。ウェブ・ブラウザは、ユーザの信用証明書をアクセス・ポイントに送信するためにハイパーテキスト転送プロトコル（ＨＴＴＰ）またはセキュア・ソケット・レイヤを介したハイパーテキスト転送プロトコル（ＨＴＴＰＳ）を使用する。ＨＴＴＰに関する問題は、パスワードがデータ接続を介して送信される前に暗号化されず、平文として送信されるということである。つまり、パスワードがハッカーによる傍受を受けやすいということである。例えば、データ接続にアクセスするハッカーは、ネットワーク監視アプリケーションを使用して、データ接続全体で送信されているデータ・パケットを捕足し表示することができる。このようなネットワーク監視アプリケーションは一般的であり、その使用が違法であるためにこれらはしばしばパケット・スニッフィング・アプリケーションまたはパケット・スヌーピング・アプリケーションと呼ばれている。ＨＴＴＰＳに関する問題は、アクセス・ポイントが良く知られた認証局（ＣＡ）から信用証明書を取得する必要があるということである。これは、アクセス・ポイントを設定する費用を上昇させる。ＨＴＴＰＳによって使用される暗号化の強度は、政府の輸出規制の制限を受ける。ウェブ・ブラウザは、ディフォルトの場合は比較的脆弱な鍵を含み、ユーザには輸出規制に応じて暗号化の強度をアップグレードすることが求められる。本明細書の目的のために、用語「接続アプリケーション」は、限定はしないが、ピアツーピア認証アレンジメント、ダイヤラー、スマート・クライアント、ブラウザ、サプリカント、スマート・カード、トークン・カード、ＰＤＡ接続アプリケーション、無線接続、組み込み型認証クライアント、イーサネット接続などの、データを認証する機能を含むいかなるデバイス（ハードウェアとソフトウェアの両方）を含むものと解釈されるべきである。
【課題を解決するための手段】
【００１７】
本発明により、少なくとも１つのサービス・プロバイダを含むサービス・アクセス・システムで生成された複数のトランザクション・データ記録を関連付ける方法であって、トランザクション・データ記録は、単一ユーザ・セッション中にユーザがシステムにアクセスしたのに応えて生成され、

単一ユーザ・セッションに一意に関連付けられ、少なくとも１つのサービス・プロバイダによって受信可能であり、トランザクション・データ記録に含まれる一意のセッションＩＤを生成するステップと、
トランザクション処理ファシリティで少なくとも１つのサービス・プロバイダから複数のトランザクション・データ記録を受信するステップと、
各トランザクション・データ記録の一意のセッションＩＤを使用してトランザクション・データ記録を処理するステップと
を含む方法が提供される。

【００１８】

さらに本発明により、少なくとも１つのサービス・プロバイダを含むサービス・アクセス・システムで生成されたトランザクション・データ記録を処理するシステムであって、トランザクション・データ記録は、単一ユーザ・セッション中にユーザがシステムにアクセスしたのに応えて生成され、
単一ユーザ・セッションに一意に関連付けられ、少なくとも１つのサービス・プロバイダによって受信可能であり、トランザクション・データ記録に含まれる一意のセッションＩＤを生成するセッションＩＤジェネレータと、
各記録の一意のセッションＩＤを使用して少なくとも１つのサービス・プロバイダから受信した複数のトランザクション・データ記録を処理するトランザクション処理ファシリティと
を含むシステムが提供される。

【００１９】

本発明のさらなる態様では、少なくとも１つのサービス・プロバイダを含むサービス・アクセス・システムで生成された複数のトランザクション・データ記録を処理する方法であって、
少なくとも１つのサービス・プロバイダからトランザクション・データ記録を受信するステップであって、各トランザクション・データ記録は、単一ユーザ・セッション中にユーザがシステムにアクセスしたのに応えて生成されるステップと、
単一ユーザ・セッションに関連付けられたトランザクション・データ記録を各トランザクション・データ記録に含まれるセッションＩＤに基づいて識別するステップであって、各セッションＩＤは単一ユーザ・セッションを一意に識別するステップと
を含む方法が提供される。

【００２０】

本発明のさらに別の態様では、少なくとも１つのサービス・プロバイダを含むサービス・アクセス・システムで生成された複数のトランザクション・データ記録を処理するトランザクション処理ファシリティであって、
少なくとも１つのサービス・プロバイダからトランザクション・データ記録を受信し、各トランザクション・データ記録は、単一ユーザ・セッション中にユーザがシステムにアクセスしたのに応えて生成され、
単一ユーザ・セッションに関連付けられたトランザクション・データ記録を各トランザクション・データ記録に含まれるセッションＩＤに基づいて識別し、各セッションＩＤは単一ユーザ・セッションを一意に識別する
ように構成されたトランザクション処理ファシリティが提供される。

【００２１】

本発明のさらに別の態様では、ユーザをアクセス・サービス・プロバイダに接続する方法であって、ユーザがサービス・プロバイダにアクセスする単一ユーザ・セッションに関連付けられた一意のセッションＩＤを作成するステップであって、ユーザ・セッションが認可される際に各トランザクション・データ記録のユーザＩＤストリングに一意のセッションＩＤが提供されるステップを含む方法が提供される。

【００２２】

本発明は、本明細書に記載の方法のどれか１つを保持する、一連の命令を組み込んだ機械

可読媒体に拡張される。
【００２３】
さらに本発明により、少なくとも１つのサービス・プロバイダを含むサービス・アクセス・システムで生成された複数のトランザクション・データ記録を処理するトランザクション処理ファシリティであって、
少なくとも１つのサービス・プロバイダからトランザクション・データ記録を受信する受信手段であって、各トランザクション・データ記録が単一ユーザ・セッション中にユーザがシステムにアクセスしたのに応えて生成される受信手段と、
単一ユーザ・セッションに関連付けられたトランザクション・データ記録を各トランザクション・データ記録に含まれるセッションＩＤに基づいて識別するプロセッサ手段であって、各セッションＩＤは単一ユーザ・セッションを一意に識別するプロセッサ手段と
を含むトランザクション処理ファシリティが提供される。
【００２４】
本発明の他の特徴および利点は、図面および以下の詳細な説明から明らかになろう。
【００２５】
本発明は、一例として説明するものであり、添付の図面の各図によって限定されることを意図するものではない。尚、添付の図面では、同様の参照は同一または類似の要素を示している。
【発明を実施するための最良の形態】
【００２６】
ネットワーク・ユーザの信用証明書またはユーザを安全に認証する方法およびシステムが開示される。ネットワーク・アクセス・デバイスは、ネットワーク・ユーザによって入力されたパスワードのようなネットワーク・ユーザの信用証明書を暗号化する。ネットワーク・アクセス・デバイスは、強力な暗号化アルゴリズムによって生成された公開／秘密鍵対の一部である公開鍵によってネットワーク・ユーザの信用証明書を暗号化する。ネットワーク・アクセス・デバイスは、暗号化されたネットワーク・パスワードをネットワーク暗号解読サーバに送信する。ネットワーク暗号解読サーバは、公開／秘密鍵対の秘密鍵を使用してネットワーク・ユーザの信用証明書を暗号解読する。ネットワーク暗号解読サーバは、暗号解読されたパスワードを検証するために認証サーバ（ＡＡＡ）に送信する。パスワードがＡＡＡサーバで肯定的に検証された場合、ＡＡＡサーバは、パスワードが適切に検証されたこと、すなわち認証されたことを示す適切な確認応答信号をネットワーク・アクセス・デバイスに送信する。この確認応答信号に基づいて、ネットワーク・アクセス・デバイスは、インターネットまたは何らかの他のソースに対するアクセスを得る。
【００２７】
ネットワーク・アクセス・デバイスで強力なアルゴリズムに基づき非対称公開鍵を使用してネットワーク・パスワードを暗号化することにより、パスワードをネットワーク・アクセス・デバイスからネットワーク暗号解読サーバに安全に送信することができる。暗号化されたパスワードが、ネットワーク・アクセス・デバイスとネットワーク暗号解読サーバの間のある地点でスニッフィング・アプリケーションまたはスヌーピング・アプリケーションによって捕足される場合、暗号化されたパスワードは正確な秘密鍵の知識と暗号化アルゴリズムによってのみ暗号解読することができる。ユーザの信用証明書の暗号解読は、ユーザがアクセスすることを希望しているソースのできる限り近くで行われることが好ましい。
【００２８】
図示した本発明の実施形態は基礎となる認証プロトコルには依存せず、したがって様々な新しい認証プロトコルと既存の認証プロトコルと共に機能するよう実現することができる。さらに本発明の実施形態は、認証チェーンの機能を完全に標準化する必要性を解決しながら、安全な認証を提供する。例えば暗号化されたデータを標準ＰＰＰ／ＲＡＤＩＵＳ情報フィールドを通過させることによって、本発明は、ＣＨＡＰやＥＡＰのようなさらに複雑な認証プロトコルと共に機能するようネットワーク機器を実装し、構成する労力と費用

を掛けずに安全な認証方法を提供する。しかし、本発明はＣＨＡＰ、ＥＡＰ、および他のプロトコルと共に使用することができ、ＰＡＰ／ＲＡＤＩＵＳ環境のアプリケーションに限定されるものではないということを理解されたい。

【００２９】

図２は、本発明の一実施形態と一致する、ＩＳＰネットワーク２５５、ネットワーク・アクセス・デバイス２０５、およびネットワーク暗号解読サーバ２４０を含むネットワーク構成２００を示す図である。ＩＳＰネットワーク２５５は、ＮＡＳ２２０、モデム・プール２１５、およびゲートウェイ２２５を含む。ＩＳＰネットワーク２５５はゲートウェイ２２５を介してインターネットに接続されており、ＮＡＳ２２０とネットワーク暗号解読サーバ２４０の間の接続を介してＩＳＰ認証システム２６５に接続されている。一実施形態では、ＩＳＰネットワーク２５５とＩＳＰ認証システム２６５は物理的に同じファシリティ内に配置されている。しかし代替形態では、ＩＳＰ認証システム２６５は１つのファシリティ内に配置されており、ワイド・エリア・ネットワーク（ＷＡＮ）を介してＩＳＰネットワーク２５５のような１つまたは複数のＩＳＰネットワークに接続されている。この種の構成は、個々のＩＳＰネットワークを異なる地理的領域内に戦略的に配置することを可能にし、したがって安全性を増すために認証システムを集中化しながら顧客はローカルな電話の呼によってネットワークにアクセスすることができる。

【００３０】

本発明の一実施形態では、インターネット２６０にアクセスするために、ネットワーク・ユーザはネットワーク・アクセス・デバイス２０５でダイヤルアップ接続アプリケーションを実行する。代替形態では、インターネットにアクセスするために他のタイプのネットワーク接続アプリケーションを利用することができる。ダイヤルアップ接続アプリケーションは、ネットワーク・ユーザ名とネットワーク・パスワードを入力し、モデム２１０を操作してモデム・プール２１５との音声通信セッションを確立するようネットワーク・ユーザを促す。図２ではモデム２１０は外部デバイスとして示してあるが、本発明の代替形態では、モデム２１０はネットワーク・アクセス・デバイス２０５に内蔵した内部デバイスであってよい。音声通信セッションが確立されると、ＮＡＳ２２０はネットワーク・ユーザを認証するためにネットワーク・アクセス・デバイス２０５との通信を開始する。

【００３１】

ネットワーク・アクセス・デバイス２０５がネットワーク・ユーザによって入力されたネットワーク信用証明書を送信する前に、ネットワーク・パスワードが暗号化される。パスワードは公開／秘密鍵対の公開鍵を使用して暗号化される。この暗号化技術は当技術分野では良く知られており、一般には非対称公開鍵暗号法と呼ばれている。非対称公開鍵暗号法では、人は１つの鍵を公的に使用可能とし、第２の秘密鍵を保持する。メッセージは公開鍵によって「ロック」すなわち暗号化され、送信され、次いで秘密鍵によって「アンロック」すなわち暗号解読される。

【００３２】

図示する本発明の本実施形態では、公開／秘密鍵対を生成するために強力な暗号化アルゴリズムが使用される。公開鍵と秘密鍵は楕円曲線に基づく数学的関係を有している。この暗号化技術は当技術分野では良く知られており、一般には楕円曲線暗号法またはＥＣＣと呼ばれている。公開鍵暗号化アルゴリズムは、秘密鍵から公開鍵を生成することは容易になるが、公開鍵が与えられて秘密鍵を推論することは困難な一方向性の数学的な問題に依存している。楕円曲線システムは、楕円曲線によって作成された母集団内の公開鍵と秘密鍵の間の関係を決定するために数式を使用する。楕円曲線暗号法は、他の強力な暗号技術と比べて鍵のサイズが小さいので有利である。これはパスワードを強力な暗号化方法で暗号化することを可能にしながら、暗号化されたパスワードは尚もＰＡＰ、ＣＨＡＰ、ＥＡＰ、およびＲＡＤＩＵＳのような普及している認証プロトコルによって定義されるパスワード・データ・フィールドに適合する。

【００３３】

再度図２を参照すると、秘密鍵は秘密鍵データベース２４５に記憶させておき、公開鍵は

ネットワーク・アクセス・デバイス２０５に知られている。ネットワーク・アクセス・デバイス２０５は、ネットワーク・ユーザ名と暗号化されたネットワーク・パスワードをＮＡＳ２２０に送信する前に公開鍵を使用してパスワードを暗号化する。ＮＡＳ２２０は、ネットワーク・ユーザ名と暗号化されたネットワーク・パスワードをネットワーク暗号解読サーバ２４０に転送する。ネットワーク暗号解読サーバ２４０は、秘密鍵データベース２４５へのインデックスとしてネットワーク・ユーザ名を使用し、ネットワーク・ユーザ名に関連付けられた秘密鍵を取り出す。次いでネットワーク暗号解読サーバ２４０は、秘密鍵を使用して、暗号化されたネットワーク・パスワードを暗号解読し、ネットワーク・ユーザによって入力されたオリジナルの平文パスワードを生成する。

【００３４】

最後に、ネットワーク暗号解読サーバ２４０は、ネットワーク・ユーザ名と平文ネットワーク・パスワードを検証するためにＡＡＡサーバ２３５に転送する。ＡＡＡサーバ２３５は、ネットワーク・ユーザ名に関連付けられた公式パスワードを取り出すために、認証データベース２３０へのインデックスとしてネットワーク・ユーザ名を使用する。公式パスワードがネットワーク・ユーザによって入力され、ネットワーク・アクセス・デバイス２０５によって送信されたパスワードと一致する場合、ＡＡＡサーバ２３５は適切な確認応答信号をＮＡＳ２２０に送信し、ＮＡＳ２２０はその信号をネットワーク・アクセス・デバイス２０５に転送し、これにより成功した検証の確認応答を行い、インターネットまたはある種の他のソースへのアクセスを認める。

【００３５】

本発明の一実施形態は、ネットワーク・アクセス・デバイス２０５からＮＡＳ２２０に、最終的にはＡＡＡサーバ２３５に信用証明書を送信するために使用される認証プロトコルには依存しない。例えば、本発明は、特にＰＡＰ、ＣＨＡＰ、ＥＡＰ、およびＲＡＤＩＵＳのような普及している認証プロトコルと共に機能するように実施することができる。

【００３６】

本発明の一実施形態に関して、ＮＡＳ２２０はネットワーク・ユーザの信用証明書を認証するためにＰＡＰおよびＲＡＤＩＵＳを使用するよう構成されている。ＰＡＰ／ＲＡＤＩＵＳ用に構成されている場合、ＮＡＳ２２０は、ＮＡＳ２２０とネットワーク・アクセス・デバイス２０５の間で通信セッションが開始される際に、ＰＡＰの使用についてネットワーク・アクセス・デバイス２０５と交渉する。ＮＡＳ２２０は、ＲＡＤＩＵＳサーバであるＡＡＡサーバ２３５のＲＡＤＩＵＳクライアントとして構成される。ネットワーク暗号解読サーバ２４０はＲＡＤＩＵＳサーバとしても構成されているが、ＡＡＡサーバ２３５に対してＲＡＤＩＵＳプロキシ・クライアントとして動作する。この構成では、ネットワーク・アクセス・デバイス２０５は、ネットワーク・ユーザによって入力された際にパスワードを暗号解読する。次いでネットワーク・アクセス・デバイス２０５はＰＡＰパケットを作成し、ネットワーク・ユーザ名と暗号化されたネットワーク・パスワードをパケット内の適切なフィールドに置く。次にネットワーク・アクセス・デバイス２０５はＰＡＰパケットをＮＡＳ２２０に送信する。ＮＡＳ２２０は、ＲＡＤＩＵＳパケットを使用してデータをネットワーク暗号解読サーバ２４０に転送する。ネットワーク暗号解読サーバ２４０はパスワードを暗号解読し、ＲＡＤＩＵＳを使用して検証のために平文パスワードをＡＡＡサーバ２３５に転送する。

【００３７】

代替形態では、ＮＡＳ２２０は、ネットワーク・ユーザの信用証明書を認証するためにＣＨＡＰとＲＡＤＩＵＳを使用するように構成される。ＣＨＡＰ／ＲＡＤＩＵＳを使用するように構成されているネットワークでは、ＮＡＳ２２０はＰＡＰではなくＣＨＡＰを認証プロトコルとして使用することをネットワーク・アクセス・デバイス２０５と交渉する。次にＮＡＳ２２０は乱数を生成し、それをネットワーク・アクセス・デバイス２０５に送信する。ネットワーク・アクセス・デバイス２０５上で実行されているダイヤルアップ接続アプリケーションは、事前設定された暗号化アルゴリズムを使用してパスワードのノンリバーシブル・ハッシュを生成するために乱数を使用する。実際のパスワードを暗号化す

るのではなく、ネットワーク・アクセス・デバイス２０５は、上記の本発明の本実施形態によりネットワーク・パスワードのノンリバーシブル・ハッシュを暗号化する。ネットワーク・アクセス・デバイス２０５はＣＨＡＰパケットを作成し、ネットワーク・ユーザ名と暗号化されたノンリバーシブル・ハッシュをＮＡＳ２２０に送信する。

【００３８】

ＮＡＳ２２０は、ネットワーク・ユーザ名、暗号化されたノンリバーシブル・ハッシュ、およびノンリバーシブル・ハッシュを生成するために使用されたオリジナルの乱数を含むデータを、ＲＡＤＩＵＳプロトコルを使用してネットワーク暗号解読サーバ２４０に送信する。ネットワーク暗号解読サーバ２４０はそのノンリバーシブル・ハッシュを暗号解読し、ＲＡＤＩＵＳパケット内のノンリバーシブル・ハッシュと置き換え、これがＡＡＡサーバ２３５に転送される。

【００３９】

ＡＡＡサーバ２３５はこのパケットを受信し、認証データベース２３０からネットワーク・ユーザ名に関連付けられたパスワードを取り出す。ＡＡＡサーバ２３５は、ＮＡＳ２２０で元々生成された乱数を使用して、認証データベース２３０から取り出されたオリジナルのパスワードに対してハッシュ・オペレーションを実行する。次にＡＡＡサーバ２３５は、それ自体が生成したハッシュを、ネットワーク・アクセス・デバイス２０５から受信したハッシュと比較する。２つのハッシュが一致する場合、検証は成功し、ＡＡＡサーバ２３５は、適切な確認応答信号をネットワーク・アクセス・デバイス２０５に送信し、インターネット２６０またはある種の他のソースへのアクセスを与える。

【００４０】

本発明の別の実施形態では、ＮＡＳ２２０はＥＡＰおよびＲＡＤＩＵＳを使用するように構成されている。ＥＡＰは、ネットワーク・アクセス・デバイス２０５に送信される乱数がＮＡＳ２２０ではなくＡＡＡサーバ２３５によって生成されることを除いて、ＣＨＡＰと略同様に機能する。本発明はいかなる認証プロトコルとでも共に機能するので、本発明は様々なネットワーク構成と共に機能するように容易に実現することができ、ＬＥＧＡＣＹシステムを使用した非常に強力な最低レベルの安全性を提供することができる。

【００４１】

図３は、本発明の一実施形態と調和する、遠隔ＩＳＰネットワーク３６５、ネットワーク・アクセス・デバイス３０５、およびネットワーク暗号解読サーバ３５０を含むネットワーク構成３００を示す図である。遠隔ＩＳＰネットワーク３６５は、ＮＡＳ３２０、モデム・プール３１５、およびゲートウェイ３２５を含む。遠隔ＩＳＰネットワーク３６５は、ゲートウェイ３２５を介してインターネット３７０に接続されており、ＮＡＳ３２０とＡＡＡサーバ３３５の間の接続を介して遠隔ＩＳＰ認証システム３７５に接続されている。遠隔ＩＳＰ認証システム３７５は、ＡＡＡサーバ３３５とネットワーク暗号解読サーバ３５０との間のＷＡＮ接続を介してローカルＩＳＰ認証システム３８０に接続されている。

【００４２】

構成３００は、ネットワーク・ユーザがネットワーク・アクセス・デバイス３０５を使用して遠隔ＩＳＰネットワーク３６５によってインターネット３７０にアクセスすることができる。ローカルＩＳＰ認証システム３８０を運営・維持管理するローカルＩＳＰは、ローカルＩＳＰのネットワーク・ユーザが、遠隔ＩＳＰによって維持管理・運営されている遠隔ＩＳＰネットワーク３６５を介してインターネットにアクセスすることができるように、遠隔ＩＳＰと協定を結ぶ。この種の業務協定は、遠隔ＩＳＰが遠隔の地理的領域またはローカルＩＳＰと異なる国家に配置されている場合に発生する場合がある。図３に示す本発明の実施形態は、ローカルＩＳＰの運営者が、遠隔ＩＳＰネットワーク３６５と遠隔ＩＳＰ認証システム３７５から構成される機器にアクセスした人物を制御することが不可能なので、この種の構成では特に有利である。さらに遠隔ＩＳＰネットワーク３６５は、パスワードの暗号化されたバージョンにだけアクセスし、これにより安全性は強化される。

【００４３】
図３に示す本発明の実施形態は、暗号化されたパスワードが遠隔ＩＳＰネットワーク３６５と遠隔ＩＳＰ認証システム３７５を通過することを除いて、図２に関して上述した方法と略同様の方法で機能する。図３を参照すると、インターネット３７０にアクセスするために、ネットワーク・ユーザは、ネットワーク・アクセス・デバイス３０５のダイヤルアップ接続アプリケーションを実行する。ダイヤルアップ接続アプリケーションは、ネットワーク・ユーザ名とネットワーク・パスワードを入力し、モデム３１０を操作してモデム・プール３１５と音声通信セッションを確立するようネットワーク・ユーザを促す。一度音声通信セッションが確立されると、ＮＡＳ３２０はネットワーク・ユーザを認証する目的でネットワーク・アクセス・デバイス３０５との通信を開始する。

【００４４】
ネットワーク・パスワードをＮＡＳ３２０に送信する前に、ネットワーク・アクセス・デバイス３０５は、上記のように公開鍵によってネットワーク・パスワードを暗号化する。次いでネットワーク・アクセス・デバイス３０５は、ローカルＩＳＰ認証システム３８０宛てのデータ・パケットを作成し、そのパケットを遠隔ＩＳＰ３６５のＮＡＳ３２０に転送する。ＮＡＳ３２０は、暗号化されたパスワードを含んでいるそのデータ・パケットを受信し、それを特に遠隔ＩＳＰ認証システム３７５とＡＡＡサーバ３３５に転送する。ＡＡＡサーバ３３５はそのデータ・パケットを検査し、それがローカルＩＳＰ認証システム３８０宛てであることを発見し、そのデータ・パケットをネットワーク暗号解読サーバ３５０に転送する。

【００４５】
ネットワーク暗号解読サーバ３５０はそのデータ・パケットを受信し、秘密鍵データベース３３５からネットワーク・ユーザ名に関連付けられた秘密鍵を取り出す。次いでネットワーク暗号解読サーバ３５０は、暗号化されたパスワードを暗号解読し、検証するためにそのデータ・パケットを平文パスワードと共にＡＡＡサーバ３４５に転送する。ＡＡＡサーバ３４５は、認証データベース３４０からユーザ名に関連付けられた平文パスワードを取り出すために、認証データベース３４０へのインデックスとしてネットワーク・ユーザ名を使用する。取り出されたパスワードが、ネットワーク・アクセス・デバイス３０５から受信したパスワードと一致する場合、ＡＡＡサーバ３４５は、適切な確認応答信号を遠隔ＩＳＰ３６５のＡＡＡサーバ３３５に送信する。ＡＡＡサーバ３３５は、その信号をＮＡＳ３２０に転送する。ＮＡＳ３２０は、その信号をネットワーク・アクセス・デバイス３０５に転送し、成功した検証を確認応答し、インターネットまたはある種の他のソースへのアクセスを認める。したがって、ユーザに関連付けられたローカルＩＳＰの近くで暗号解読が行われ、暗号化された認証データにアクセスするのは任意の１つまたは複数の中間ＩＳＰだけである。

【００４６】
図４は、本発明の一実施形態と調和する、ネットワーク・ユーザの信用証明書を安全に認証する方法の動作４００の流れ図である。この方法は動作４０５から開始される。動作４０５で、ネットワーク・アクセス・デバイスは、公開／秘密鍵対の一部である公開鍵を使用してパスワードのようなネットワーク信用証明書を暗号化する。公開／秘密鍵対は、楕円曲線暗号法のような強力な暗号化アルゴリズムに基づいて前もって生成されている。

【００４７】
動作４１０で、ネットワーク・アクセス・デバイスは暗号化されたネットワーク信用証明書をネットワーク暗号解読サーバに送信する。暗号化されたパスワードは、最終的にネットワーク暗号解読サーバに到達する前にネットワーク・アクセス・サーバとＡＡＡサーバを含む複数のネットワーク・ノードを介して転送される。

【００４８】
動作４１５で、ネットワーク暗号解読サーバは、上記の公開／秘密鍵対の秘密鍵を使用して暗号化されたネットワーク信用証明書を暗号解読する。ネットワーク暗号解読サーバは、ユーザ名を秘密鍵データベースへのインデックスとして使用して秘密鍵データベースか

ら秘密鍵を取り出す。
【００４９】
最後に、動作４２０で、ネットワーク暗号解読サーバは、暗号解読されたネットワーク信用証明書を検証するためにＡＡＡサーバに送信する。最終的に検証のためＡＡＡサーバに到達する前に、ネットワーク・アクセス・サーバまたは他のＡＡＡサーバのようないくつかのネットワーク・ノードを介して暗号解読されたネットワーク信用証明書を転送することができる。
【００５０】
本発明の典型的な適用例はマルチパーティ・サービス・アクセス環境であるが、そのアプリケーションについては後述する。このような適用例は、通常、ローミング・ユーザ、複数のサービス・プロバイダ、および複数の顧客を含む。さらに、このような適用例は、通常は、ＰＡＰ、ＣＨＡＰ、ＥＡＰ、ＲＡＤＩＵＳ、またはユーザの信用証明書を安全でない方法で通信する類似のプロトコルを使用する。しかし、後述する実施形態は、ＬＥＧＡＣＹシステムでの安全な認証を可能にする。
【００５１】
用語
本明細書の目的で、「サービス・アクセス・トランザクション」という用語は、ユーザ・セッションに関するサービス顧客とサービス・プロバイダとの間のいかなるトランザクションをも含んでいる。このようなサービスの一例として、任意の媒体またはプロトコルを介した任意の通信ネットワークへのアクセスをあげることができる。例えば通信ネットワークは、パケット交換ネットワーク、回線交換ネットワーク、ケーブル・ネットワーク、衛星ネットワーク、地上ネットワーク、有線ネットワーク、または無線ネットワークを含むことができる。しかし「サービス・アクセス・トランザクション」という用語は、ネットワーク・アクセス・トランザクションに限定されるものではなく、コンテンツ・サービス、商取引サービス、および通信サービスのような多数の他のサービスの任意の１つへのアクセスに関するトランザクションを含むものである。
【００５２】
本発明の目的のために、「顧客」という用語は、サービス・アクセスが顧客によって行われるか否かに関わらず、サービス・アクセスの購入および／または消費に関与する任意のエンティティを含む。例えば「顧客」は、実際にサービス・アクセスを利用するエンド・ユーザ消費者、またはそのようなエンド・ユーザが属している企業体、インターネット・サービス・プロバイダ、インターネット通信事業者、再販業者、またはチャネルであってよい。
【００５３】
マルチパーティ・サービス・アクセス環境
本発明の本実施形態は、サービス・プロバイダ（例えばＩＳＰ、無線サービス・プロバイダ、ＶＰＮサービス・プロバイダ、コンテンツ配信サービス・プロバイダ、電子商取引サービス・プロバイダ、またはアプリケーション・サービス・プロバイダ）が標準通信プロトコル（例えばＰＰＰ、ＨＴＴＰ）および標準認証プロトコル（例えばＲＡＤＩＵＳ、ＰＡＰ、ＥＡＰなど）を使用してマルチパーティ・アクセス環境で比較的安全なサービス・アクセスを提供できるようにする、サービス・アクセス（例えばインターネット・アクセス、コンテンツ・アクセス、商取引アクセス、または通信アクセス）サービス用のマルチパーティ・アクセス・ブローカー／設定システムを開示する。このようなプロトコルは、通常、最大長のユーザ・フィールドを規定し、本発明の実施形態は、特に上記の最大長のフィールド内で安全な認証を提供する方法およびシステムを記述する。したがって、本発明はＬＥＧＡＣＹシステムに適用することができる。
【００５４】
概要
図５は、多数のサービス・プロバイダ４５２、アクセス・ブローカー・システム４５４、および複数の顧客（または消費者）４５６を含むネットワーク・アクセス環境の一形態で

あるマルチパーティ・サービス・アクセス環境例４５０のブロック図である。高レベルでは、サービス・プロバイダ４５２は、アクセス・ブローカー・システム４５４を介して複数の顧客４５６に販売されるサービス（例えばアクセス・サービス、コンテンツ・サービス、電子商取引サービスなどの）・キャパシティを有する。したがって、アクセス・ブローカー・システム４５４は、顧客４５６に転売されるサービス・キャパシティ（例えばサービス・アクセス）を購入すると見なすことができる。サービスへのアクセスはネットワーク・アクセスとして後述されているが、アクセスはサービスの一例として後述されており、本明細書の目的では上記のアクセスのどの形態でも含むものと見るべきであるということが理解されよう。本実施形態では、サービス・プロバイダ４５２は、ＩＳＰ４５８（例えばＵＵＮｅｔ技術、Ｇｅｎｕｉｔｙ、ＣｏｍｐｕＳｅｒｖｅ Ｎｅｔｗｏｒｋ Ｓｅｒｖｉｃｅｓ、ＥＱＵＡＮＴ、Ｈｏｎｇ Ｋｏｎｇ Ｔｅｌｅｃｏｍなど）、無線アクセス・プロバイダ４６０（例えばＶｅｒｉｚｏｎ、Ｓｐｒｉｎｔ、Ｐａｃｉｆｉｃ Ｂｅｌｌ）、コンテンツ配信プロバイダ４６２、および電子商取引プロバイダ４６４のような任意の通信ネットワーク・サービス・プロバイダを含むことができる。しかしサービス・プロバイダ４５２は、任意の数のサービス（数例をあげるならば、例えばアクセス・サービス、コンテンツ・サービス、通信サービス、または電子商取引サービス）を提供する任意の種類のサービス・プロバイダをいくつでも含むことができる。

【００５５】

アクセス・ブローカー・システム例４５４は多数の構成要素を含む。接続アプリケーションは、通常、ダイヤルアップ・アプリケーションまたは接続ダイヤラー４６６の形態で、通信ネットワークへの便利なアクセスに役立つ顧客４５６のサービスまたはネットワーク・アクセス・デバイス（例えば図２および３のアクセス・デバイス２０５、３０５のようなコンピュータ・システム）にインストールされているクライアント・アプリケーションである。一実施形態では、接続ダイヤラー４６６は、アクセス・ブローカー・システム４５４の世界規模の接続ネットワークへのダイヤリングのために簡単なポイント・アンド・クリック・インターフェースを提供することができる。この目的のために、接続ダイヤラー４６６は、潜在的に異なる設定とダイヤルアップ・スクリプト記述情報と共に世界中の複数のＩＳＰに対する複数の電話番号を記憶することができる。図１から４に関して上記で概説したように、ポイント・ツー・ポイント・プロトコル（ＰＰＰ）、パスワード認証プロトコル（ＰＡＰ）、チャレンジ・ハンドシェーク認証プロトコル（ＣＨＡＰ）、遠隔認証ダイヤルイン・ユーザ・サービス（ＲＡＤＩＵＳ）プロトコル、ターミナル・アクセス・コントローラ・アクセス制御システム（ＴＡＣＡＣＳ）プロトコル、ライトウェイト・ディレクトリ・アクセス・プロトコル（ＬＤＡＰ）、ＮＴドメイン認証プロトコル、Ｕｎｉｘ（登録商標）パスワード認証プロトコル、ハイパーテキスト転送プロトコル（ＨＴＴＰ）、セキュア・ソケット・レイヤを介したハイパーテキスト転送プロトコル（ＨＴＴＰＳ）、拡張認証プロトコル（ＥＡＰ）、転送レイヤ・セキュリティ（ＴＬＳ）プロトコル、トークン・リング・プロトコルおよび／またはセキュア遠隔パスワード・プロトコル（ＳＲＰ）のような周知のプロトコルによって許可または可能とされるユーザ・ストリングにユーザ・データとカウンター・データを含むことができるような方法で接続ダイヤラー４６６はユーザ・データとカウンター・データを暗号化する。

【００５６】

環境４５０は、ユーザ識別情報、認証応答および使用法、およびアカウンティング情報を経路指定し、ロギングする信頼あるサードパーティの機能を提供する複数のトランザクション・サーバ４６８も含む。トランザクション・サーバ４６８は、暗号解読機能を含むので、暗号解読サーバを規定することができる。

【００５７】

接続ダイヤラー４６６がクライアントまたはユーザのネットワーク・アクセス・デバイス２０５、３０５にインストールされるのに対し、ネットサーバ４７０はローミング・ユーザにＰＯＰを利用することを許可している「遠隔」ＩＳＰにインストールされ、またローム・サーバ４７２は関連付けられたホーム・ネットワークにローム・ユーザがアクセスす

ることを許可するために「ホーム」ＩＳＰに常駐する。トランザクション・サーバ４６８は、ネットワークとローム・サーバ４７０と４７２の間でメッセージを経路指定するように動作するということに留意されたい。

【００５８】

柔軟な価格設定エンジン４７６を含む設定システム４７４は、サービス・プロバイダ４５２と顧客４５６の間でサービス・アクセス・トランザクションの金銭上の決済を実行する。アクセス・ブローカー・システム４５４は、顧客４５６に提供されるサービスに関するサービス品質（ＱｏＳ）情報の収集・分析に役立つサービス品質モニタ４７８（ＳＱＭ）と顧客４５６が使用する複数の接続ダイヤラー４６６の管理に役立つ電話帳管理システム４８０も含む。トランザクション・サーバ４６８は、トランザクション・データをロードするために設定システム４７４によってアクセスされる。環境４５０の様々な構成要素は、周知の機能の態様を含むことができ、本発明の特定の実施形態によってはある種の構成要素を省略することができる。

【００５９】

顧客

図示する実施形態で顧客４５６はマルチティア顧客構造で構成されており、これによってアクセス・ブローカー・システム４５４は様々な事業計画とニーズに従って運営する顧客４５６と対話することができる。この範囲の一端で、顧客４５６はアクセス・ブローカー・システム４５４を介してローミング・システムに加入している個々のエンド・ユーザを含むことができる。別法として、顧客４５６は、企業の従業員のためにローミング・インターネット・アクセスを購入する法人顧客４８２（例えば法人または企業）の形態であってよい。

【００６０】

各顧客４５６は、それ自体の顧客（例えばエンド・ユーザ４８６および法人顧客４８２）に転売するためにローミング・インターネット・アクセスを購入するＩＳＰ顧客４８４を含むこともできる。各顧客４５６は、ソリューション・パートナーまたはアクセス・ブローカー・システム４５４によって仲介されるローミング・インターネット・アクセスを市場に出しエンド・ユーザ４８６、法人顧客４８２、および／またはＩＳＰ顧客４８４に転売する再販業者４８８として運営することができる。

【００６１】

顧客４５６は、インターネット通信事業者４９０（例えばＩＸＣ、ＲＢＯ、ＣＬＥＣ、ＩＬＥＣ、およびＩＳＰ）と見なされる関係者を含むこともできる。したがって、マルチパーティ・アクセス環境４５０では、多数の異なるサービス・プロバイダが、ローミング・ユーザにアクセスを提供することに参加することができ、したがって顧客のセキュリティ問題と、特にユーザの安全な認証が重要な課題になるということが理解されよう。また、参加者数の増加に伴い、アカウンティングの問題はより複雑化する傾向がある。

【００６２】

ローミング・サービス・アクセス

特に図６を参照すると、参照番号５００は、アクセス・ブローカー・システム４５４が本発明の一実施形態によって比較的安全な方式でローミング・インターネット・アクセスを提供することができる方法の一例を全般的に示している。「ホーム」ＩＳＰ５０４の加入者として示されているローミング・ユーザ５０２が、特定の地理的領域５１０内のローカルＰＯＰ５０８を提供する遠隔ＩＳＰ５０６に接続すると、ローミング・ユーザ５０２は「ホーム」ＩＳＰ５０４のＰＯＰ５０９を介して接続している際に使用されるのと同じユーザ名５１２とパスワード５１４（すなわち、認証データまたはユーザの信用証明書）を入力する。しかし、標準またはＬＥＧＡＣＹマルチパーティ・アクセス環境は、通常、ダイヤルアップ認証にはＰＡＰを、有線および無線ブロードバンド認証にはＨＴＴＰ　ＰＯＳＴベースの認証を使用する。この結果、パスワードは安全でない媒体を介して転送されることになり、そのパスワードの機密性は損なわれ、それ以後、アクセス・ブローカー・システム４５４および顧客４５６の両方のネットワークに不正にアクセスするために使用

される可能性がある。この問題を改善するために、本発明の一実施形態により、上記で図１から４を参照して説明し、図５から１３を参照してマルチパーティ環境について説明したように、接続ダイヤラー４６６はユーザのデータをＰＯＰ５０８に通信する前にそれを暗号化する。

【００６３】

図示する実施形態では、顧客４５６は接続ダイヤラー４６６を要求するためにウェブ形式を使用する。このウェブ形式は、必要となるカスタマイゼーションを指定するために使用することができるフィールドを含む。例えば次のフィールドは、平文形式のセキュアード・パスワード認証（以下では「セキュアＰＡＰ」と称する）用のウェブ形式に含まれる。

セキュアＰＡＰ暗号化をイネーブルする：（Ｙ／Ｎ）

公開鍵：＊＊＊＊

鍵ＩＤ：（０－９）

【００６４】

顧客４５６がローミング・ユーザ５０２に対するセキュアＰＡＰをイネーブルするよう希望する場合（図６を参照のこと）、かれらは自分たちの関連付けられたローム・サーバ４７２に含まれるＥＣＣユーティリティを使用する。ローム・サーバ４７２は、アクセス・ブローカー・システム４５４から顧客４５６に提供されたアプリケーションを実行し、公開／秘密鍵対を生成する。秘密鍵は通常、ｅｓｐ＿ｋｅｙ＿ｐａｉｒ．ｔｘｔファイルに追加される。公開鍵は通常、適切な形式を使用してアクセス・ブローカー・システム４５４のダイヤラー・サポート・チームに送信される。ダイヤラー・サポート・チームは、本発明の一実施形態により接続ダイヤラー４６６を構築するためにダイヤラー・カスタマイゼーション・ツール（ＤＣＴ）を使用する。ＤＣＴツールは、使用されるべき暗号化／暗号解読アルゴリズムとＥＣＣ公開／秘密鍵を指定するためのウェブ・ページを含む。

【００６５】

接続ダイヤラー４６６は、ユーザ・アクセス・セッションを一意に識別する一意のセッションｉｄ（ブロック５２２を参照のこと）を生成するために、ダイヤラーｉｄとカウンター（図７のブロック５２０を参照のこと）を維持する。接続ダイヤラー４６６は、例えばアクセス・ブローカー・システム４５４のウェブ・サーバからダイヤラーｉｄを取得し、使用時には、接続ダイヤラー４６６は各ユーザ・アクセス・セッションが一意に識別されるようダイヤル試行ごとにカウンターを増分する。ダイヤラーｉｄとカウンターの値は、一意のセッションｉｄプレフィックスを作成するために使用される。ダイヤラーｉｄとカウンターの値またはカウント（平文で送信される）の整合性を保証するために、これらのフィールドはチェックサム文字を生成するために使用される。このチェックサム文字を生成するために使用されるアルゴリズムは、以下で図１２を参照してより詳細に説明される。一意のセッションｉｄの実施形態については、本明細書でより詳細に後述される。

【００６６】

ネットサーバ４７０は、認証されたユーザｉｄとパスワードのキャッシュを限られた期間だけ維持し、したがってそれ以後の認証はそのキャッシュから実行することができる（ブロック５３８を参照のこと）。安全なＰＡＰはユーザｉｄとパスワードを認証ごとに変更するので、それはネットサーバ４７０でいかなるキャッシング機能をもブレークする。したがって、特定の実施形態では、標準化されたネットサーバのキャッシュとの整合性を維持するために、ダイヤラー４６６は限られた期間だけランダム・ポイントをローカルに記憶し、これを再利用することができる（ブロック５４０を参照のこと）。上記処理の後で、ネットサーバ４７０は認証データをトランザクション・サーバ４６８に通信する。

【００６７】

特に図８を参照すると、参照番号５５０は、トランザクション・サーバ４６８によって実行される機能例を全般的に示している。トランザクション・サーバ４６８は、ダイヤラーｉｄ、カウンターで最後に使用された値、およびテーブルでの最後のアクセス・タイムを維持する（ブロック５５２を参照のこと）。テーブルは、ネットワークをリプレイ攻撃から保護するために使用される。このテーブルは、通常、すべてのトランザクション・サー

バ４６８全体で複製される。

【００６８】

ユーザの信用証明書または認証データをネットサーバ４７０から受信する際、本発明の一実施形態では、トランザクション・サーバ４６８はパスワードを暗号解読し、受信したカウンターの値をデータベースに記憶されている値と比較する（ブロック５５４を参照のこと）。ダイヤラー４６６によって送信されるカウントがデータベースに記憶されている最後のカウント値よりも大きい場合、それは本物の要求であると見なされる（ブロック５５６を参照のこと）。ダイヤラー４６６によって送信されたカウントがデータベースに記憶されている最後のカウント値と等しく、現在のシステム・タイムとデータベースに記憶されている最後のアクセス・タイムとの間のデルタすなわち時間差が許可されたタイム・ウィンドウよりも少ない場合、ここでもまた要求は本物と見なされる（ブロック５５８を参照のこと）。ダイヤラー４６６によって送信されたカウントがこれら２つの条件を満たさない場合、トランザクション・サーバ４６８は認証要求を起こりうるリプレイ攻撃として拒否する（ブロック５６０を参照のこと）。トランザクション・サーバ４６８は、平文パスワードと共に認証要求を図９のローム・サーバ４７２に送信する。

【００６９】

図８に示す実施形態では、トランザクション・サーバ４６８は顧客の秘密鍵の記録を維持し、したがって認証データの暗号解読は暗号解読サーバを規定することのできるトランザクション・サーバ４６８で行われる。しかし特定の顧客が、トランザクション・サーバ４６８のようないかなる仲介者にも自分の秘密鍵を提供することを希望しない場合がある。このような場合、顧客の秘密鍵はトランザクション・サーバ４６８には提供されないが、一般に企業内の位置にある顧客のローム・サーバ４７２には提供される。したがって、上記に加えて、または上記の代わりに、認証データの暗号解読は顧客のローム・サーバ４７２で上記の方法と同様の方法で行うことができる。暗号化機能を含むローム・サーバ４７２の一実施形態を図９に示す。

【００７０】

特に図９を参照すると、参照番号５７０は、ローム・サーバ４７２によって実行される機能例を全般的に示す。この機能は図８の機能５５０に略類似しているので、同一または類似の機能を示すためには類似の参照番号を使用した。トランザクション・サーバ４６８が特定の顧客の秘密鍵にアクセスしない場合、トランザクション・サーバ４６８は必須のＥＣＣ属性を認証要求パケットに追加し、それをローム・サーバ４７２に送信する（ブロック５７２を参照のこと）。ローム・サーバ４７２は、ローカルに記憶されているＥＣＣ情報と秘密鍵を使用してパスワードとチェックサム文字を暗号解読する（ブロック５５２を参照のこと）。次いでローム・サーバ４７２は、カウントが有効であるか否かを判定するために上記と同様の機能テストを実行する（ブロック５５４～５６０を参照のこと）。トランザクション・サーバ４６８がそのデータベースをカウントの最新値で更新することができるように（ブロック５７６参照のこと）、ローム・サーバ４７２は、暗号解読されたカウントを認証返信パケットに追加する（ブロック５７４を参照のこと）。カウンター機能を実施するためのテーブルの一例を以下に示す。

【００７１】

ｄｉａｌｅｒ＿ｃｏｕｎｔｅｒ＿ｔｓテーブルは、通常は複製用に使用される。このテーブルは各トランザクション・サーバ４６８で作成される。

【００７２】

【表１】

表：DIALER_COUNTER_TS	
フィールド名	説明
DIALER_COUNTER_TS_ID	オラクルのスナップショットに 要求される数値ID
SERVER_ID	トランザクション・サーバID VARCHAR2(20)
DIALER_ID	ダイヤラーIDは、システム454の ウェブ・サーバでDIALER_IDサーブレット から取得される　VARCHAR2(10)
COUNTER	カウンターの最後に使用された値 VARCHAR2(5)
ACCESS_TIME	最後のアクセス・タイム

【００７３】
最後に使用された値は、通常、例えば「ＳＥＳＳＩＯＮ」マシン上のデータベース・インスタンスに記憶される。ＳＥＳＳＩＯＮマシンは、通常、トランザクション・サーバ４６８のｄｉａｌｅｒ＿ｃｏｕｎｔｅｒ＿ｔｓテーブルからエントリをプルし、それらを単一テーブルに蓄積するために使用される。ＳＥＳＳＩＯＮマシンは、トランザクション・サーバ４６８内のすべてのｄｉａｌｅｒ＿ｃｏｕｎｔｅｒ＿ｔｓテーブルに対応する一意のスナップショットの作成もする。これらのスナップショットは、通常、ｄｉａｌｅｒ＿ｃｏｕｎｔｅｒ＿ｔｓ＜ＳｅｒｖｅｒＩｄ＞と呼ばれる。ここでＳｅｒｖｅｒＩｄは、特定のトランザクション・サーバ４６８のｉｄである。データベース・インスタンスの一例であるＳＥＳＳＩＯＮは、故障耐性を強化するために両方の沿岸で２つの同一マシンにより作成される。
【００７４】
【表２】

表：DIALER_COUNTER	
フィールド名	説明
DIALER_ID	ダイヤラーIDは、システム454のシステム・ ウェブ・サーバでDIALER_IDサーブレットか ら取得され、この記録を一意に識別するため に使用される　VARCHAR2(10)
COUNTER	カウンターの最後に使用された値 VARCHAR2(5)
ACCESS_TIME	最後のアクセス・タイム

【００７５】
各トランザクション・サーバ４６８は、通常、Ｏｒａｃｌｅのスナップショットを使用してｄｉａｌｅｒ＿ｃｏｕｎｔｅｒテーブルを複製する。本発明の本実施形態に適応するように標準システムがアップグレードされる場合、一般に以下の変更例が作成される。
【００７６】
【表３】

表：SECURE_PAP	
フィールド名	説明
SPAP_ID	この記録を一意に識別した生成されたID
CUSTOMER_ID	顧客ID
PUBLIC_KEY	公開鍵
PRIVATE_KEY	秘密鍵の値
KEY_VERSION	鍵のバージョン番号
ALGORITHM	アルゴリズム。例えばEおよびA
EXPIRATION_DATE	この記録が期限切れになる時刻／日付。ヌルの場合、この記録は期限切れにはならない
DESCRIPTION	DCTから入力された説明
CREATION_DATE	記録が作成された時刻／日付
MODIFY_BY	記録を変更したユーザ
MODIFY_TIME	記録が変更された時刻

【００７７】
【表４】

表：CUSTOMER	
フィールド名	説明
ENCRYPT_FLAG	0=暗号化は任意選択、 1=この顧客には暗号化が必要

【００７８】
【表５】

表：DIALER_PROFILE	
フィールド名	説明
ENCRYPT_FLAG	0=暗号化オフ、1=暗号化オン
SPAP_ID	SECURE_PAPテーブルを参照する

【００７９】
暗号化／暗号解読機能
図７から９を参照して上記で説明した本実施形態では、ダイヤラー４６６、トランザクション・サーバ４６８、およびローム・サーバ４７２は、ＥＣＣアルゴリズムを実施し、パスワードの暗号化と暗号解読のためにＡＰＩを提供するＥＣＣ　ＡＰＩを含む。通常、ＥＣＣ実施態様は、暗号化／暗号解読のために最適な正規基底の数学を使用する。ある種の実施形態では、数学的反転の回数を１回の乗算の犠牲に縮小するように、多項式基底の数学と最適な正規基底の数学が組み合わされる。
【００８０】
特に図１０を参照すると、参照番号５８０は、ダイヤラー４６６の暗号化機能の一例を全般的に示す。ブロック５８２で示すように、暗号化アルゴリズムはＥＣＣ曲線上にランダムな点を生成する。次いでこのランダムな点は、ＥＣＣストリングの一部＜暗号化されたパスワード＞を作るためにパスワードとチェックサム文字（ブロック５８４を参照のこと

）を暗号化するために使用される。ダイヤラー４６６はランダムな点を暗号化し、それをネットサーバ４７０に送信する（ブロック５８６および５８７を参照のこと）。通常、この暗号化には後述するように記号変換方式が使用される。

【００８１】

既存のプロトコル、例えばＰＰＰ、ＰＡＰ、ＲＡＤＩＵＳなどに対応するために、パスワード・フィールドは印刷可能なＵＳ－ＡＳＣＩＩ文字を有する。ある種の実施形態では、文字はＲＦＣ２４８６標準に準拠するような方法で生成される。これらの実施形態では、パスワードおよびチェックサム・フィールドが暗号化される際、標準プロトコルを使用しているネットワークに適用することができるようにストリングを許容可能な文字で生成するよう配慮がなされる（ブロック５８８を参照のこと）。したがって、この符号化を実行するために以下の文字変換方式を使用することができる。符号化されるべき各文字は、まず以下に示すテーブルに従う値にマッピングされる。

【００８２】

【表６】

#	記号	#	記号	#	記号	#	記号
0.	0	1.	1	2.	2	3.	3
4.	4	5.	5	6.	6	7.	7
8.	8	9.	9	10.	A	11.	B
12.	C	13.	D	14.	E	15.	F
16.	G	17.	H	18.	I	19.	J
20.	K	21.	L	22.	M	23.	N
24.	O	25.	P	26.	Q	27.	R
28.	S	29.	T	30.	U	31.	V
32.	W	33.	X	34.	Y	35.	Z
36.	A	37.	B	38.	C	39.	D
40.	E	41.	F	42.	G	43.	H
44.	I	45.	J	46.	K	47.	L
48.	M	49.	N	50.	O	51.	P
52.	Q	53.	R	54.	S	55.	T
56.	U	57.	V	58.	W	59.	X
60.	Y	61.	Z	62.	~ (チルダ)	63.	` (アクサン グレーブ)
64.	! (エクスクラメーション・マーク)	65.	# (番号記号)	66.	$ (ドル記号)	67.	% (パーセント記号)
68.	^ (カレット)	69.	& (アンド記号)	70.	* (星印)	71.	((左括弧)
72.	) (右括弧))	73.	- (ハイフン またはマイナス)	74.	_ (下線)	75.	+ (プラス記号)
76.	= (等号)	77.	{ (左中括弧)	78.	[(左大括弧)	79.	} (右中括弧)
80.	] (右大括弧)	81.	| (縦線)	82.	\ (バック スラッシュ)	83.	: (コロン)
84.	; (セミコロン)	85.	" (コーテーション・マーク)	86.	' (アポストロフィ)	87.	< (小なり記号)

#	記号	#	記号	#	記号	#	記号
			MARK)				SIGN)
88.	, （コンマ）	89.	> （大なり記号）	90.	? （クエスチョン・マーク）	91.	（スペース）
92.	/ （スラッシュ）	93.	. （終止符）	94.	@ （単価記号）		

【００８３】
マッピングされた値は次いでランダムな点の対応するバイトに加えられ、モデュラス９５が計算される（ブロック５９０を参照のこと）。この結果、文字は上記テーブルの別の文字にマッピングされる。暗号解読サーバで文字を暗号解読するには、符号化された文字からランダムな点の対応するバイトが引かれ（図１１のブロック５８１を参照のこと）、結果のモデュラス９５が計算される（ブロック５８３を参照のこと）。結果が負の数である場合、オリジナルの文字を得るためにその結果に値９５が加えられる（ブロック５８５を参照のこと）。一例として、「ｒ」を符号化に使用されるランダムな点のバイトとし、「ｘ」をオリジナルの文字として、
符号化：ｙ＝（ｘ＋ｒ）％９５
復号：ｘ＝（ｙ－ｒ）％９５
（ｘ＜０）の場合、
ｘ＝ｘ＋９５である。
【００８４】
ダイヤラー４６６での暗号化プロセス中に、パスワード・フィールドとチェックサム文字はランダムな点で暗号化される。これらのフィールドのそれぞれは、符号化のためにランダムな点の異なるバイトの組を使用する。パスワード・フィールドは、その符号化のために第１のバイトの組を使用し、チェックサム・フィールドはその符号化のためにバイト１０を使用する。
【００８５】
ダイヤラーｉｄとカウンター値の整合性を確認するためにチェックサム文字が使用される。ダイヤラーｉｄとカウンター値が平文で送信される場合、悪意のある人物は、これらの値を変更して、リプレイ攻撃に対する保護を破ることができる。この問題に対処するため、チェックサム文字がダイヤラーｉｄとカウンター値から生成され、その後、それはランダムな点を使用して符号化される（図１２のブロック５９２を参照のこと）。暗号化されたチェクサム文字は次いで、ユーザｉｄストリングの一部として送信される。
【００８６】
チェックサム文字は、カウント値、ダイヤラーｉｄ、およびランダムな点のＭＤ５ハッシュによって生成される（図１２のブロック５９２および５９４を参照のこと）。次いで７ビットがハッシュから選択され、上記の符号化方法を使用してランダムな点からの単一バイトによって符号化される（バイト＃１０）（ブロック５９６を参照のこと）。符号化されたビットは次いで、暗号化された点の最後の７バイトに分散され（ブロック５９８を参照のこと）、ユーザのストリングの一部として送信される（ブロック５９９を参照のこと）。ダイヤラー４６６が符号化されたデータをトランザクション・サーバ４６８またはローム・サーバ４７２に送信する際、場合によっては、これらはチェックサムを独立して生成することによって（図８および９のブロック５８８を参照のこと）ダイヤラーｉｄとカウンター値を検証し、それをダイヤラー４６６から送信されたチェックサムと比較し（ブ

ロック５９０を参照のこと）、それらが一致しない場合には拒絶する。
【００８７】
図１０のダイヤラー４６６に戻ると、符号化されたストリングは次いで以下のように連結されてＥＣＣストリングを形成する。
＜符号化されたパスワード＞＜最後の７バイトにおける符号化されたチェックサム・ビットを有するランダムな点の暗号化され符合化されたｘ座標＞
【００８８】
その後、ダイヤラー４６６はＥＣＣストリングをダイヤラーｉｄとカウンター値に連結し、それをＰＡＰなどのプロトコルのユーザｉｄとパスワード・フィールドで送信する。例えば＜符号化されたパスワード＞＜最後の７バイトにおける符号化されたチェックサム・ビットを有するランダムな点の暗号化され符号化されたｘ座標＞＜ダイヤラーｉｄ＞＜カウンター値＞。
【００８９】
図１０で説明した方法は、そのようなストリング長を有する暗号化されたストリングを作り、暗号化されたストリングをＬＥＧＡＣＹシステムを使用して通信することができるような性質を特徴として含むということに留意されたい。
【００９０】
暗号化論理は、通常、以下の署名によってｉｐ＿ｓｐａｐ＿暗号（）メソッドにカプセル化される。
ｃｈａｒ＊ ｉｐ＿ｓｐａｐ＿ｅｎｃｒｙｐｔ（ｃｏｎｓｔ ｃｈａｒ＊ａｌｇｏｒｉｔｈｍ、ｃｏｎｓｔ ｃｈａｒ ｐｕｂｌｉｃ＿ｋｅｙ、ｃｏｎｓｔ ｃｈａｒ ｐａｓｓｗｏｒｄ、ｃｏｎｓｔ ｃｈａｒ＊ｄｉａｌａｒ＿ｉｄ、ｃｏｎｓｔ ｃｈａｒ＊ｃｏｕｎｔｅｒ、ｃｈａｒ＊＊ｐｌａｉｎ＿ｐｏｉｎｔ、ｃｈａｒ＊＊ｅｎｃｒｙｐｔｅｄ＿ｐｏｉｎｔ、ｉｎｔ＊ｒｅｔｕｒｎＣｏｄｅ）
ここで、
ａｌｇｏｒｉｔｈｍは使用されるべきアルゴリズムである。ＳｅｃｕｒｅＰＡＰ ｐｕｂｌｉｃ＿ｋｅｙの「Ｓ」はＥＣＣ公開鍵（ｃｏｎｆｉｇ．ｉｎｉより）であり、
ｐａｓｓｗｏｒｄは平文パスワードであり、
ｄｉａｌｅｒ＿ｉｄはダイヤラーのｉｄであり（ダイヤラーｉｄサーブレットから取得）、
ｃｏｕｔｎｅｒはダイヤル試行カウント（ダイヤル試行ごとにダイヤラーによって増分される）であり、
ｐｌａｉｎ＿ｐｏｉｎｔ－このフィールドが空のままである場合は、新しいランダムな点が生成される。このフィールドは、戻る際に符号化するために使用されるランダムな点を指し示す。
ｅｎｃｒｙｐｔｅｄ＿ｐｏｉｎｔ－このフィールドが空のままである場合は、暗号化された点を生成するためにプレーンな点と公開鍵が使用される。このフィールドは、戻る際にメソッドが使用する暗号化された点を指し示す。
戻りコード０－呼が成功した場合、非ゼロ・コードが提供される。このメソッドは、成功の場合にはＥＣＣストリングを戻し、そうでない場合にはヌルを戻す。
【００９１】
暗号解読論理はｉｐ＿ｓｐａｐ＿暗号解読（）メソッドにカプセル化される。このメソッドは以下の署名を有する。
ｃｈａｒ＊ ｉｐ＿ｓｐａｐ＿ｄｅｃｒｙｐｔ（ｃｏｎｓｔ ｃｈａｒ＊ａｌｇｏｒｉｔｈｍ、ｃｏｎｓｔ ｃｈａｒ ｐｒｉｖａｔｅ＿ｋｅｙ、ｃｏｎｓｔ ｃｈａｒ ｅｃｃ＿ｓｔｒｉｎｇ、ｃｏｎｓｔ ｃｈａｒ＊ｄｉａｌａｒ＿ｉｄ、ｃｏｎｓｔ ｃｈａｒ＊ｃｏｕｎｔｅｒ、ｉｎｔ＊ｒｅｔｕｒｎＣｏｄｅ）、ここで、
ａｌｇｏｒｉｔｈｍは使用されるべきアルゴリズムである。ＳｅｃｕｒｅＰＡＰ ｐｒｉｖａｔｅ＿ｋｅｙの「Ｓ」はＥＣＣ秘密鍵（ｓｅｃｕｒｅｐａｐテーブルまたはｅｓｐ＿ｋｅｙ＿ｐａｉｒ．ｔｘｔファイルより）であり、

ｅｃｃ＿ｓｔｒｉｎｇは暗号（）メソッドによって戻されるストリングであり、
ｄｉａｌｅｒ＿ｉｄはダイヤラーのｉｄ（ダイヤラーｉｄサーブレットから取得）であり、
ｃｏｕｎｔｅｒはダイヤル試行カウント（ダイヤル試行ごとにダイヤラーによって増分される）であり、
戻りコード０－呼が成功した場合であり、そうでない場合は非ゼロ・コードである。
【００９２】
このメソッドは、成功の場合には平文パスワードを戻し、そうでない場合にはヌルを戻す。
【００９３】
ダイヤラーのカスタマイゼーション形式
上述のように、顧客４５６は、ＳｅｃｕｒｅＰＡＰを使用して通信するよう構成されたカスタマイズされたダイヤラーを要求するためにウェブ形式を使用する。このウェブ形式は、通常、必要なカスタマイゼーションを指定するために使用することのできるフィールドを含んでいる。このウェブ形式は、以下のフィールドの例を含むことができる。
ＳｅｃｕｒｅＰＡＰ暗号化をイネーブルする：（Ｙ／Ｎ）
公開鍵：
鍵Ｉｄ：（０－９）
【００９４】
ダイヤラーのカスタマイゼーション・ツール
カスタマイゼーションプロセス中、アクセス・ブローカー・システム４５４の管理者は、ＳｅｃｕｒｅＰＡＰを使用するダイヤラー４６６を生成するという選択肢を有している。イネーブルされた場合、以下のフィールドの例を、通常はダイヤラー４６６にパッケージされているｃｏｎｆｉｇ．ｉｎｉに設定することができる。
［ｉＰａｓｓなどの処理機能ＩＤ］
暗号化フラグ＝ｙｅｓ
アルゴリズム＝Ｓ
鍵のバージョン＝０
公開鍵＝ＢｗＡＡＡＭＧｄｑＹｘ２１ｘｈＷｔＥＱＭｄＤＨｈｗｖＵ＝＆ＡＱＡＡＡＦｄｄ４０ｕＬＱＭＤ１ＵＴｔｙＢｑＤＨＹ＝
【００９５】
特定の顧客４５６の対応するダイヤラー４６６から送信されたパスワードをトランザクション・サーバ４６８が暗号解読できるように、これらの値はトランザクション・データベースにも記憶されている。本実施形態では、このファイルには公開鍵しか記憶されておらず、秘密鍵は暗号化の安全のために秘密にされる。
【００９６】
ＳｅｃｕｒｅＰＡＰをイネーブルすることに加えて、カスタマイゼーション・ツールは、使用されるアルゴリズムと鍵のバージョンを設定するという選択肢も提供する。例えば、以下の暗号化アルゴリズムをサポートすることができる。
暗号化しない場合にはＡ
楕円曲線暗号化の場合にはＥ
一意のセッションＩＤに適合するＥＣＣの場合にはＳ
一意のセッションＩＤの場合にはＵ
【００９７】
実際には、Ａは主としてテストおよびデバッグのためのものである。Ｅは、ダイヤラーがダイヤラーｉｄを有しない場合にパスワードを暗号化するために使用される。Ｕは、暗号化アルゴリズムではないが、後で詳述するように一意のセッションｉｄを識別するために使用される。鍵のバージョンは０から開始されるが、既存のダイヤラー・プロファイルに新しい鍵対が求められるたびに増分される。ダイヤラー４６６は、ｓｅｃｕｒｅ＿ｐａｐテーブルにＥＣＣ鍵および他の情報を記憶する。次いでこのテーブルはＯｒａｃｌｅのス

ナップショットを介してトランザクション・サーバ４６８に複製される。秘密鍵が漏洩すると、新しい鍵対が生成される。秘密鍵の安全性が損なわれると、ダイヤラー・サポート・チームは以下の行動を取るべきである。
１．漏洩した鍵に対して適切な失効期日を設定する。漏洩した鍵を使用しているすべてのダイヤラー４６６がその鍵をもう１回だけ使用することができるよう保証するにはこれで十分であるはずである。ダイヤラー４６６は古い鍵を使用してインターネットに接続し、アップデート・サーバから新しい鍵を有するｃｏｎｆｉｇ．ｉｎｉファイルを取り出す。顧客４５６がパスワードを暗号解読するためにローム・サーバ４７２を使用している場合、顧客４５６は、通常、失効期日以後はｅｓｐ＿ｋｅｙ＿ｐａｉｒ．ｔｘｔファイルから漏洩した鍵を手動で除去する。
２．新しい鍵対を生成するか、または新しい鍵対を生成するよう顧客４５６に要求し、公開鍵をアクセス・ブローカー・システム４５４に送信する。
３．ＤＣＴツールを使用し、公開鍵を置き換える（新しい鍵ｉｄを使用して）。ダイヤラーを構築する。
【００９８】
ダイヤラー
ダイヤラー４６６は、パスワードを暗号化すべきか否かを判定するためにｃｏｎｆｉｇ．ｉｎｉファイルをチェックする。ＳｅｃｕｒｅＰＡＰがイネーブルされている場合、ダイヤラー４６６はｃｏｎｆｉｇ．ｉｎｉファイルの公開鍵を使用し、ｉｐ＿ｓｐａｐ＿暗号化（）メソッドを呼び出すことによってパスワードを暗号化する。このメソッドはＥＣＣストリングを作成し、これを戻す。ダイヤラー４６６はＥＣＣストリングをダイヤラーｉｄとカウンター値に連結する。ＥＣＣストリングの最初の１６文字がパスワード・フィールドに置かれ、ストリング内の残りの文字はプレフィックス・フィールド（０Ｓまたは０Ｅプレフィックスと共に）に置かれる。ダイヤラー４６６はダイヤラーｉｄを取得するまではアルゴリズム「Ｅ」を使用する。このプレフィックスは、すべてのシステムおよび経路指定プレフィックスの後、顧客のプレフィックスの前に含まれる。ダイヤルされているＰＯＰが電話帳内のＰＡＰプレフィックスに適合しないプレフィックスを有している場合、ダイヤラー４６６はパスワードを暗号化せず、ＳｅｃｕｒｅＰＡＰプレフィックスを作成しない。暗号化プレフィックスを含むサンプルのユーザ名を以下に示す。
ユーザＩＤ：ＩＰＡＳＳ／０Ｓ　Ａｘｒｔ５０ｚＴｘｃａ５４６ｈｊｄｇｋｂｘｃｊｃ？＿ｄ０ｗｅ／ｊｏｅ@ｉｐａｓｓ．ｃｏｍ
パスワード：ｘ３５～！４Ｑｕ｛ｘｙ７１］Ｄ８
ここで、鍵のバージョン＝０でありアルゴリズム＝Ｓである。
【００９９】
アクセス・ブローカー・システム４５４は、暗号化が必要ないと判定した場合、ダイヤラー・カウントから一意のセッションｉｄを作成し、それをプレフィックス・フィールドに置く。一意のセッションｉｄプレフィックスを含むサンプルのユーザ名を以下に示す。
ユーザＩＤ：ＩＰＡＳＳ／０ＵＡｘｒｔ５ＡＢ２／ｊｏｅ@ｉｐａｓｓ．ｃｏｍ
パスワード：ｔｈｉｓｉｓａｂｉｇｓｅｃｒｅｔ
ここで、鍵のバージョン＝０であり、アルゴリズム＝Ｕである。
ダイヤラー４６６はそのローカル・ストレージにｐｌａｉｎ＿ｐｏｉｎｔと暗号化された点を記憶する。
【０１００】
リダイヤルが試行された場合、ダイヤラー４６６はカウンターを増分し、プレーンな点と暗号化された点を使用してｉｐ＿ｓｐｓｐ＿暗号化（）メソッドを呼び出す。
【０１０１】
顧客ソリューション
顧客ソリューション・プロセスは［０－９］［Ａ－Ｚ］＊／形式のプレフィックスの有無をチェックする。そのようなプレフィックスが発見されず、顧客４５６がパスワードの暗号解読を要求しない場合、顧客ソリューションは正常に動作する。プレフィックスが発見

された場合、最初のスラッシュ（／）までの後ろから８バイトが取り除かれ、一意のセッションｉｄフィールドとして記憶される。顧客ソリューションコードは、０Ｓ＜ｄｉａｌｅｒ＿ｉｄ＞＜ｃｏｕｔｅｒ＞というコンテンツを有する一意のセッションｉｄフィールドを作成することができる。整数が取り除かれ、鍵識別フィールドとして記憶される。アルゴリズムが取り除かれ、別個のフィールドとして記憶される。

【０１０２】

ダイヤラー・カウンターの複製

図示する安全なＰＡＰの実施形態は、リプレイ攻撃に対する保護のためにｄｉａｌｅｒ＿ｃｏｕｎｔｅｒテーブルを使用する。各トランザクション・サーバ・データベースは、ｄｉａｌｅｒ＿ｃｏｕｎｅｒ＿ｔｓテーブルを含んでいる。トランザクション・サーバ４６８は、ＳｅｃｕｒｅＰＡＰプレフィックスを有する正常な認証要求を受信するか否かに関わらず、新しい行をこのテーブルに挿入する。この行の内容は、ｓｅｒｖｅｒ＿ｉｄ、ｄｉａｌｅｒ＿ｉｄ、カウンターおよびシステム・タイム（ＧＭＴで）を含む。

【０１０３】

ＳＥＳＳＩＯＮデータベースは、すべてのトランザクション・サーバ４６８でのｄｉａｌｅｒ＿ｃｏｕｎｔｅｒ＿ｔｓテーブルに対するスナップショットを含んでいる。これらのスナップショットは、通常、ｄｉａｌｅｒ＿ｃｏｕｎｔｅｒ＿ｔｓ＜ＳＥＲＶＥＲ＿ＩＤ＞と呼ばれる。ここで、＜ＳＥＲＶＥＲ＿ＩＤ＞は特定のトランザクション・サーバ４６８のｉｄである。

【０１０４】

トランザクション・サーバ４６８からのスナップショットをリフレッシュするために「リフレッシュ」ツールが提供される。ｄｉａｌｅｒ＿ｃｏｕｎｔｅｒ＿ｔｓ＿＜ＳＥＲＶＥＲ＿ＩＤ＞は、挿入されているカウンターの値がｄｉａｌｅｒ＿ｃｏｕｎｔｅｒテーブルのカウンターの値と等しいかまたはそれよりも大きい場合、挿入される行のｄｉａｌｅｒ＿ｉｄ、カウンター、およびａｃｃｅｓｓ＿ｔｉｍｅをｄｉａｌｅｒ＿ｃｏｕｎｔｅｒテーブルに更新／挿入する「ＯＮ　ＩＮＳＥＲＴ」ＰＬ／ＳＱＬトリガーを有する。トランザクション・サーバ４６８は、ＳＥＳＳＩＯＮデータベースのｄｉａｌｅｒ＿ｃｏｕｎｔｅｒスナップショットをリフレッシュするためにリフレッシュ・ツールを使用する。次いでｄｉａｌｅｒ＿ｃｏｕｎｔｅｒテーブルは、より高速なアクセスのためにトランザクション・サーバ４６８によりキャッシュされる。実行中にｄｉａｌｅｒ＿ｃｏｕｎｔｅｒテーブルの記録に何らかの変更が加えられた場合、その影響は即座に現れる。これは、データベース・トリガーとｃｈａｃｈｅ＿ｕｐｄａｔｅテーブルを使用するアクセス・ブローカー・システム４５４の他の構成要素で使用されているのと同じ機構を使用することによって達成される。

【０１０５】

トランザクション・サーバ

起動時、効率的なルックアップのためにアクセス・ブローカー・システム４５４は、データベースのすべての秘密鍵をローカル・キャッシュに読み込む。これは、特定の顧客４５６がパスワードの暗号化を要求しているか否かを示すために顧客のキャッシュに追加の属性も有する。トランザクション・サーバ４６８は、ｄｉａｌｅｒ＿ｃｏｕｎｔｅｒテーブルもキャッシュする。実行中にこれらのテーブルに何らかの変更が加えられた場合、その影響は即座に現れる。これは、データベース・トリガーとｃｈａｃｈｅ＿ｕｐｄａｔｅテーブルを使用するアクセス・ブローカー・システム４５４の他の構成要素で使用されているのと同じ機構を使用することによって達成される。

【０１０６】

暗号化されたプレフィックス・フィールドが「Ｓ」アルゴリズムを指定する場合、トランザクション・サーバ４６８はパスワード・フィールドの内容を、顧客ソリューション・プロセスによって構築された暗号化されたプレフィックス・フィールドに連結し、「ＥＣＣフィールド」を作成する。ＥＣＣフィールドは以下を含んでいる。

＜符号化されたパスワード＞＜ランダムな点の暗号化され符号化されたｘ座標＞＜符号化

されたチェックサム文字＞
【０１０７】
トランザクション・サーバ４６８は、鍵インデックスを使用して適切な顧客４５６に対する秘密鍵を探し出す。秘密鍵がデータベースで発見された場合、パスワードを暗号解読し復号するためにトランザクション・サーバ４６８はｉｐ＿ｓｐａｐ＿暗号解読（）メソッドを呼び出す。次いでパスワード・フィールドはローム・サーバ４７２に送信される前に平文パスワードによって上書きされる。
【０１０８】
秘密鍵がキャッシュで発見されない場合、トランザクション・サーバ４６８は、通常、認証要求パケットに以下のフィールドを付加し、それをローム・サーバ４７２に送信する。そのフィールドとはすなわち、アルゴリズム、鍵インデックス、ＥＣＣフィールド（パスワードとして）、ダイヤラーｉｄ、カウンター、最後に使用されたカウンターの値およびアクセス・タイム（データベースから）、および「ｙｅｓ」に設定された「ｄｅｃｒｐｔ＿ａｔ＿ｒｏａｍＳｅｒｖｅｒ」フラグである。
【０１０９】
次いでトランザクション・サーバ４６８は、認証の細目をｉｐ＿ａｕｔｈ＿ｔｒａｎｓテーブルに、ｄｉａｌｅｒ＿ｃｏｕｎｔｅｒの細目をｄｉａｌｅｒ＿ｃｏｕｎｔｅｒ＿ｔｓテーブルに記憶する。トランザクション・サーバ４６８は、通常、挿入が更新よりも一般に高速の場合にはいつでも新しいｄｉａｌｅｒ＿ｃｏｕｎｔｅｒ＿ｔｓ記録を挿入する。
【０１１０】
トランザクション・サーバ４６８は、アカウント要求を受信すると、一意のセッションｉｄを作成するために顧客ソリューション・プロセスを使用し、それをパケットに「ｉｐａｓｓ＿ｓｅｓｓｉｏｎ＿ｉｄ」として付加する。ｔｒ＿ｕｓｅｒｉｄフィールドは、ｒａｗ＿ｕｓｅｒｉｄを含んでいる。
【０１１１】
ＥＳＰツール
公開／秘密鍵対を生成し、暗号化アルゴリズムと暗号解読アルゴリズムをテストするために、顧客４５６はそのローム・サーバ４７２、ＤＣＴチーム、およびＱＡチームと共にＥＳＰコマンド行ツールを使用する。
ｅｓｐ＿ｇｅｎｋｅｙ（ローム・サーバ４７２を有する顧客４５６の場合）
【０１１２】
このツールは、ｅｓｐ＿ｋｅｙ＿ｐａｉｒ．ｔｘｔと呼ばれるファイルに公開／秘密ＥＳＰ鍵対をプリントする。このファイルは、Ｕｎｉｘの／ｕｓｒ／ｉｐａｓｓ／ｋｅｙｓディレクトリとＷｉｎｄｏｗｓ（登録商標）用のＩＰＡＳＳ＿ＨＯＭＥ／ｋｅｙｓディレクトリに常駐する。ダイヤラー４６６を公開鍵で構築することができるように、鍵はアクセス・ブローカー・システム４５４に、例えば安全なウェブサイトを介しても提出される。通常、秘密鍵の安全なバックアップも維持される。
ｅｓｐ＿ｇｅｎｋｅｙ＿ｄｃｔ：
【０１１３】
このツールは、公開／秘密ＥＳＰ鍵対を標準出力にプリントする。これはＤＣＴの要件を満たす形式でプリントされる。出力例を以下に示す。
1
公開
鍵：ＢｇＡＹＶＫ１ａｚＵｔ８ｃｏｍｋ４１ＧｚＬｗ＝＆ＡＤＩｋＧｆＭｇＮＣｈＭ４ｖＹ６＋ｎＬｇＴｑｏ＝
秘密鍵：ＡＱＡＡＡＡＺＯＳＮＨ１３ＰａＧ３ＮｕｑＧｂＵ７ＴＹ０＝
【０１１４】
最初の行は、鍵の生成が成功したことを示す「１」を含んでいる。エラーが発生した場合、出力は「０」の値を有する。
【０１１５】

ｅｓｐ＿ｑａ：
このツールは、ＥＣＣ　ＡＰＩをテストするために使用可能な複数のコマンド行の選択肢を有している。サポートされる選択肢のサンプルの一例を以下に示す。
ｅｓｐ＿ｑａ　ｇｅｎｋｅｙ
ｅｓｐ＿ｑａ　ｅｎｃｒｙｐｔ［－ａ＜ａｌｇｏｒｉｔｈｍ＞－ｄ＜ｄｉａｌｅｒ＿ｉｄ＞－ｃ＜ｃｏｕｎｔｅｒ＞］－ｋ＜ｐｕｂｌｉｃ＿ｋｅｙ＞－ｔ＜ｔｅｘｔ＞
ｅｓｐ＿ｑａ　ｄｅｃｒｙｐｔ［－ａ＜ａｌｇｏｒｉｔｈｍ＞－ｄ＜ｄｉａｌｅｒ＿ｉｄ＞－ｃ＜ｃｏｕｎｔｅｒ＞］－ｋ＜ｐｒｉｖａｔｅ＿ｋｅｙ＞－ｔ＜ｔｅｘｔ＞
ｅｓｐ＿ｑａ　ｔｅｓｔｉｐｇ［－ａ＜ａｌｇｏｒｉｔｈｍ＞－ｄ＜ｄｉａｌｅｒ＿ｉｄ＞－ｃ＜ｃｏｕｎｔｅｒ＞］－ｋ＜ｐｕｂｌｉｃ＿ｋｅｙ＞－ｔ＜ｔｅｘｔ＞－ｕ＜ｕｉｄ
＠ｄｏｍａｉｎ＞
ｅｓｐ＿ｑａ　ｔｅｓｔ　－ｔ＜ｔｅｘｔ＞
括弧［］内の選択肢は任意選択である。各ｅｓｐ＿ｑａコマンドは以下のように記述される。
ｇｅｎｋｅｙ－公開／秘密鍵対を生成する。
ｅｎｃｒｙｐｔ－所与のｐｕｂｌｉｃ＿ｋｅｙによってテキスト（パスワード）を暗号化する。
ｄｅｃｒｙｐｔ－所与の秘密鍵によってテキスト（パスワード）を暗号解読する。
ｔｅｓｔｉｐｇ－「暗号化」を実行し、次いで所与のユーザに対してｃｈｅｃｋ－ｉｐｅｎコマンドを実行する。
ｔｅｓｔ－基本的なＥＣＣ　ＡＰＩテスト。アルゴリズムＳに対してｇｅｎｋｅｙ、ｅｎｃｒｙｐｔ、およびｄｅｃｒｙｐｔを実行する。
【０１１６】
ローム・サーバ
ローム・サーバ４７２は、通常、トランザクション・サーバ４６８から受信したパケットの「ｄｅｃｒｙｐｔ＿ａｔ＿ｒｏａｍｓｅｒｖｅｒ」フィールドの有無をチェックする。このフィールドがある場合、ローム・サーバはパケットの「鍵インデックス」フィールドを使用し、ｅｓｐ＿ｋｅｙ＿ｐａｉｒ．ｔｘｔファイルの秘密鍵をフェッチする。秘密鍵、ダイヤラーｉｄ、およびカウンター値を有するＥＣＣストリングが、ｉｐ＿ｓｐａｐ＿暗号解読（）メソッドに送られる。ｉｐ＿ｓｐａｐ＿暗号解読（）メソッドは、このパスワードを復号し暗号解読する。次いでユーザを認証するためにローム・サーバ４７２によって平文パスワードが使用される。
【０１１７】
図６に戻り、ダイヤラー４６６が一度上記の方法を実行すると、認証データは、遠隔ＩＳＰ５０６の認証サーバ６００に送信された後でＮＡＳ５３２に通信される。動作の正規のコースでは、遠隔ＩＳＰ５０６のＮＡＳ５３２は提供された認証情報を拒絶する。しかし図６に示すように、ネットサーバ４７０は、この認証情報を正規のユーザ要求ではなくローミング・ユーザの認証要求として認識することを容易にするために認証情報を代行受信する。
【０１１８】
認証サーバ６００は、ネットサーバ４７０と共に、ローミング・ユーザ５０２に関連付けられたローミング・ドメイン名または経路指定プレフィックスを決定するために、受信した認証情報を構文解析する。そのようなドメイン名またはプレフィックスが存在する場合、ユーザの認証情報は上記のように暗号化され、ネットサーバ４７０からセキュア・ソケット・レイヤ（ＳＳＬ）を介してトランザクション・サーバ４６８に送信される。
【０１１９】
トランザクション・サーバ４６８は、要求を経路指定するためにセッションＩＤの顧客経路指定プレフィックスを使用することができる。代わりに、トランザクション・サーバ４６８は、インターネット・プロトコル（ＩＰ）ルックアップを実行することができ、認証

要求を適切なホームＩＳＰ５０４に経路指定する。より具体的には、トランザクション・サーバ４６８は、遠隔ＩＳＰ５０２のネットサーバ４７０から暗号化された認証要求を受信し、この要求を図７から９を参照して上記で説明したように暗号解読する。次いでトランザクション・サーバ４６８は、所望のホームＩＳＰ５０４のローミング・ドメイン名または経路指定プレフィックスを参加者ドメイン名およびＩＰアドレスの現行リストと照合することによって「ホーム」ＩＳＰ５０４を決定する。照合が成功した場合、認証要求は暗号化され、ホームＩＳＰ５０４に常駐しているローム・サーバ４７２にＳＳＬを介して送信される。識別されたローム・サーバ４７２が指定期間内に応答しない場合、トランザクション・サーバ４６８は、関連ドメインのＩＳＰの代替ローム・サーバ４７２に接触するよう試みる。

【０１２０】

次いでホームＩＳＰ５０４のローム・サーバ４７２は、上記のようにトランザクション・サーバ４６８から送信された認証要求を暗号解読し、その認証要求をホームＩＳＰの正規の認証サーバ６０２に、あたかもそれが顧客プレフィックスを使用するホームＩＳＰ５０４の所有するターミナル・サーバまたはＮＡＳ５３２であるかのように提出する。ホームＩＳＰ５０４の認証サーバ６０２は、その要求に応えて、認証要求に含まれるユーザ名およびパスワードの有効性に基づいて「アクセス許可」または「アクセス拒否」応答を提供する（図８を参照のこと）。ホームＩＳＰの認証サーバ６０２からの応答は、ローム・サーバ４７２によって受信され、暗号化され、トランザクション・サーバ４６８に送信されて戻される。

【０１２１】

一意のセッションＩＤ

複数のトランザクション・データ記録を関連付ける方法およびシステムの一例を以下に示す。この方法およびシステムは、通常上記の暗号化／暗号解読方法と共に使用される一意のセッションｉｄの生成および使用を記載する。

【０１２２】

上記のように、ポイント・ツー・ポイント（ＰＰＰ）、パスワード認証プロトコル（ＰＡＰ）、チャレンジ・ハンドシェーク認証プロトコル（ＣＨＡＰ）、遠隔認証ダイヤルイン・ユーザ・サービス（ＲＡＤＩＵＳ）プロトコル、ターミナル・アクセス・コントローラ・アクセス制御システム（ＴＡＣＡＣＳ）プロトコル、ライトウェイト・ディレクトリ・アクセス・プロトコル（ＬＤＡＰ）、ＮＴドメイン認証プロトコル、Ｕｎｉｘパスワード認証プロトコル、ハイパーテキスト転送プロトコル（ＨＴＴＰ）、セキュア・ソケット・レイヤを介したハイパーテキスト転送プロトコル（ＨＴＴＰＳ）、拡張認証プロトコル（ＥＡＰ）、転送レイヤ・セキュリティ（ＴＬＳ）プロトコル、トークン・リング・プロトコル、およびセキュア遠隔パスワード・プロトコル（ＳＲＰ）などのような通信プロトコルは、ユーザＩＤストリングに備える。それぞれの異なるプロトコルが許可する文字のサイズまたは長さは異なる場合があるが、上記に列挙したプロトコル例がサポートするサイズの最小共通分母は通常は約６３文字である。この場合、一意のユーザ・セッションＩＤを提供することによって、認証、アカウンティング、およびＳＱＭ処理が強化される。

【０１２３】

例示のマルチパーティ・サービス・アクセス環境４５０に対する上記プロトコルの適用にはユーザＩＤストリングが含まれおり、したがってこれは様々な参加者が生成したすべての関連するトランザクション・データ記録、例えばトランザクション・サーバ４６８、サービス・プロバイダ４５２、および顧客４５６に共通している。しかしある種の環境では、それらのプロトコルで使用されている従来技術のユーザＩＤストリングはマルチパーティ・サービス・アクセス環境４５０の特定のユーザに一意に関連付けることができるが、特定の単一ユーザ・セッションには一意に関連付けられない。例えばネットワーク・タイムアウトおよびパケット再試行アルゴリズムが原因で、単一トランザクション・データ記録がトランザクション・サーバ４６８に複数回送信され、それらの記録のどれか１つまたは複数が不良である場合、同一の単一ユーザ・セッションに関連する記録の複数のインス

タンスが設定システム４７４に存在する場合がしばしばある。さらに、認知された失敗した通信の試行を再送する試みにおいて、ある種のＮＡＳ４７０（図６を参照のこと）は実際にユーザ・セッションＩＤストリングを変更し、その結果、同一の単一ユーザ・セッションに対して異なるトランザクション・データ記録が生じることになる。上記説明は、不十分なアカウンティング記録の２つの例にすぎないが、多数の他の状況があるということが理解されよう。

【０１２４】

本発明の別の実施形態によれば、単一ユーザ・セッションに応答して生成された関連トランザクション・データ記録は、共通の一意のセッションＩＤを含む。場合によっては、このセッションＩＤは、個々のユーザの使用情報の強力だが必ずしも絶対的ではないＩＤを提供する場合があり、この一意のユーザ・セッションＩＤは少なくともある種のパラメータ内では一意であるべきである。例えば、所与の期間中に生成されたすべての記録を、一意のユーザ・セッションＩＤを使用して関連付け、処理することができるように、一意のユーザＩＤはその所与の期間だけは一意であってよい。

【０１２５】

通常、上記のプロトコル例の場合、ユーザＩＤストリングは、ネットワークにアクセスしているユーザのユーザ名とパスワードだけでなく、顧客領域を含む経路指定情報を含む。例示のマルチパーティ・サービス・アクセス環境４５０で使用されるユーザＩＤまたはＩＤストリングは、通常は以下に示す通りである。

＜ＦａｃｉｌｉｔｙＲｏｕｔｉｎｇＰｒｅｆｉｘ＞／［＜ＦａｃｉｌｉｔｙＬｏｃａｔｉｏｎＰｒｅｆｉｘ＞］／［＜ＣｕｓｔｏｍｅｒＲｏｕｔｉｎｇＰｒｅｆｉｘ＞］／［ＣｕｓｔｏｒｍｅｒＰｒｅｆｉｘ（ｓ）］／＜ＥｎｄＵｓｅｒＮａｍｅ＞@［＜ＮｏｎＲｏｕｔｉｎｇＣｕｓｔｏｍｅｒＤｏｍａｉｎ＞］｜［＜ＣｕｓｔｏｍｅｒＲｏｕｔｉｎｇＤｏｍａｉｎ＞］

ここで、

＜ＦａｃｌｉｔｙＲｏｕｔｉｎｇＰｒｅｆｉｘ＞は、トラフィックをアクセス・ブローカー・システムまたはファシリティ４５４のネットワークに経路指定するためにＩＳＰ４５８、無線アクセス・プロバイダ４６０、コンテンツ配信プロバイダ４６２、電子商取引プロバイダ４６４など（アクセス・プロバイダ）が使用する独自開発のプレフィックスである。

＜ＦａｃｉｌｉｔｙＬｏｃａｔｉｏｎＰｒｅｆｉｘ＞は、ファシリティ４５４にアクセスを提供する点またはノードの位置を決定するためにファシリティが使用するプレフィックスである。

＜ＣｕｓｔｏｍｅｒＲｏｕｔｉｎｇＰｒｅｆｉｘ＞は、トラフィックを顧客のサイトに経路指定するためにアクセスまたはサービス・プロバイダ４５２が使用するプレフィックスである。

＜ＣｕｓｔｏｒｍｅｒＰｒｅｆｉｘ（ｓ）＞は、内部経路指定のために顧客４５６が使用するプレフィックスである。

＜ＥｎｄＵｓｅｒＮａｍｅ＞は、ファシリティ４５４を使用しているエンド・ユーザ５０２のログイン・ユーザ名である。

＜ＣｕｓｔｏｍｅｒＲｏｕｔｉｎｇＤｏｍａｉｎ＞は、トラフィックを顧客のサイトに経路指定するためにシステム４５４が使用するドメインである。ユーザＩＤストリングは＜ＣｕｓｔｏｍｅｒＲｏｕｔｉｎｇＰｒｅｆｉｘ＞または＜ＣｕｓｔｏｍｅｒＲｏｕｔｉｎｇＤｏｍａｉｎ＞を含む。

＜ＮｏｎＲｏｕｔｉｎｇＣｕｓｔｏｍｅｒＤｏｍａｉｎ＞は、内部経路指定のために顧客４５６が使用するドメインである。

【０１２６】

次に、一意のセッションＩＤを上記プロトコルの１つのプロトコルのユーザＩＤフィールドに適合させる実現可能な方法の一例を説明する。しかし一意のセッションＩＤを含めることを他の方法で実施することができるということが理解されよう。ダイヤラーがパスワ

ードの暗号化にＥタイプのアルゴリズムを使用する場合、代わりの解決法の例が実施される。Ｅタイプのアルゴリズムは、ユーザ名の暗号化されたランダムな点を含む。暗号化されたランダムな点は、個々のユーザ・セッションの強力だが必ずしも絶対的ではないＩＤを提供し、したがって一意のセッションｉｄとして使用される。

【０１２７】

上記のように、例示のプロトコルがサポートする独自開発の情報に対する最小共通分母の使用可能なストリング長は、通常、約６３文字である。一意のセッションｉｄはユーザ名フィールドが科す制限内に適合すべきである。

【０１２８】

一意のセッションＩＤを生成するために（図２０のブロック８０２を参照のこと）、各サービス・プロバイダ４５２に常駐する接続ダイヤラーの一形態である接続アプリケーション４６６は、アクセス・ブローカー・システム４５４のウェブ・サーバ８０６のサーブレットから接続アプリケーション４６６を識別するダイヤラーＩＤを取得する（図１５を参照のこと）。ダイヤラーＩＤは、通常、一意のダイヤラーＩＤでもある。ダイヤラーＩＤは、ユーザ優先ファイルに記憶されており、ダイヤラーが最初にインストールされた際には、ユーザ優先ファイルのダイヤラーＩＤは通常は空である。ダイヤラー４６６が最初に、例えばインターネットに接続した時点では、ダイヤラー４６６は、通常、ウェブ・サーバ８０６に新しいダイヤラーＩＤを要求する（ブロック８００を参照のこと）。図示する実施形態では、ダイヤラーは、ウェブ・サーバ８０６から一意のダイヤラーＩＤを取得するまでは一意のセッションＩＤを作成しない。したがって、ダイヤラーＩＤが一意のセッションＩＤの一部を構成している本実施形態では、ダイヤラー４６６からの最初の正常なセッションは一意のセッションＩＤを含んでいない。しかしダイヤラー４６６は、いかなる後続の試行に対してもそのダイヤラーＩＤを有する。

【０１２９】

ダイヤラー４６６は、固有のダイヤラーＩＤに加え、内部的に維持され、ユーザ優先ファイルに記憶されているカウンター４６７も含む。カウンター４６７はダイヤル試行のたびに増分される（ブロック８０２を参照のこと）。ダイヤラー４６６は、ダイヤラーＩＤとカウンターを使用して本実施形態では１１文字で定義される（図１８を参照のこと）セッションＩＤインジケーターを後続のダイヤル試行のたびに生成する。カウンター４６７がダイヤル試行のたびに増分されるので、ダイヤラーは大域的に一意のセッションＩＤである＜ｄｉａｌｅｒ　ｉｄ＞＜ｃｏｕｎｔｅｒ＞を生成する（ブロック８０２を参照のこと）。本実施形態では、セッションＩＤには、識別子、例えばユーザＩＤストリングがローミング・サーバ４７２に渡される前にトランザクション・サーバ４６８によって取り除かれた、ファシリティまたはアクセス・ブローカー・システム４５４に関連付けられた「ＯＵ」などの３文字によってプレフィックスが付けられる（図５を参照のこと）。したがって、一意のセッションＩＤに１１文字が含まれる場合、８文字はダイヤラーＩＤとカウンター用に使用可能であるということになる。

【０１３０】

例示のダイヤラーＩＤとカウンターの両方が基数６４を有する数を使用する。この番号付け方式に使用される記号はＡ～Ｚ、ａ～ｚ、０～９、＆および？を含む。カウンター４６７は、各ダイヤルの試行以前に増分され、ダイヤラーＩＤは0で事前充填され、本実施形態では5桁の入力によって定義される。したがって、カウンター４６７用には3桁が残っている。したがって、ダイヤラーＩＤ用に使用される5桁は１０７３７４１８２４（十億以上）の一意のダイヤラーのインストールを可能にし、3桁のカウンターは２６２１４４のダイヤル試行を可能にする（一日あたり２０の試行と仮定して、カウンターは２３年後にリセットされることになっている）。したがってこの期間中、セッションＩＤは各ユーザ・セッションを一意に定義することになる。しかし、一意のセッションＩＤに割り振られ、または使用される文字数は、システムが対応するプロトコルの１つまたは複数のタイプによってシステムごとに異なる場合があるということを理解されたい。

【０１３１】

トランザクション記録の処理
図１４は、アクセス・ブローカー・システム４５４が役立つ場合がある、本発明の一実施形態によるアカウンティングおよび設定手順を示すブロック図である。
【０１３２】
ローミング・ユーザ５０２が遠隔ＩＳＰ５０６に接続する際、セッションを管理しているターミナル・サーバ（またはＮＡＳ）４７０は、ユーザＩＤストリングを含むトランザクション・データ記録、つまり１１文字の一意のセッションＩＤを生成し、この情報を認証サーバ６００に送信する。認証サーバ６００は、ネットサーバ４７０と共にローミング・ユーザに関連付けられたローミング・ドメイン名とプレフィックスを決定するためにアカウンティング情報を構文解析する。そのようなドメイン名またはプレフィックスがある場合、ユーザのアカウンティング情報はＲＳＡデータ・セキュリティーズからのアルゴリズムを使用して暗号化され、ネットサーバ４７０からセキュア・ソケット・レイヤ（ＳＳＬ）を介してトランザクション・サーバ４６８に送信される。
【０１３３】
ローミング・ユーザ５０２が遠隔ＩＳＰ５０６から切断する際、セッションを管理しているターミナル・サーバ（またはＮＡＳ）４７０は、ユーザＩＤストリングを含むトランザクション・データ記録、つまり１１文字の一意のセッションＩＤを生成し、この情報を認証サーバ６００に送信する。認証サーバ６００は、ネットサーバ４７０と共にローミング・ユーザに関連付けられたローミング・ドメイン名とプレフィックスを決定するためにアカウンティング情報を構文解析する。そのようなドメイン名とプレフィックスがある場合、ユーザのアカウンティング情報はＲＳＡデータ・セキュリティーズからのアルゴリズムを使用して暗号化され、ネットサーバ４７０からセキュア・ソケット・レイヤ（ＳＳＬ）を介してトランザクション・サーバ４６８に送信される。
【０１３４】
トランザクション・データまたはアカウンティング記録は、アカウンティング情報がデータベースに記憶されている場合に、ＳＳＬを利用しているトランザクション・サーバ４６８に略リアルタイムで通信される。マルチパーティ・サービス・アクセス環境４５０のすべての様々な構成要素または参加者は、ユーザＩＤストリング、つまり一意のセッションＩＤを受信するが、これはトランザクション・データ記録が設定システム４７６に送信される際に、単一ユーザ・セッションに関連付けられたトランザクション・データ記録に付随する。したがって、様々な異なる参加者から送信されたトランザクション・データ記録は、それら自体が発生する元になった単一ユーザ・セッションを識別する識別子を含む。
【０１３５】
これらのアカウンティング情報は、通話詳細記録（ＣＤＲ）を作るために設定システム４７６によってさらに処理される。各通話詳細記録は、関連サービス・アクセスが行われた場合にローミング・ユーザ５０２の識別に関する詳細な使用報告と、サービス・アクセスの位置と、各サービス・アクセス・セッションの長さおよびコストと、サービス・アクセスの時間（例えばローカル時間またはＧＭＴ時間）とを提供する。
【０１３６】
複数のトランザクション・サーバ４６８は、アカウンティングまたはトランザクション・データ記録を設定システム４７６に提供し、設定システム４７６はこれらの記録を利用して顧客４５６に対する請求書（またはインボイス）を作成し、サービス・プロバイダ５０４への支払いを行う。しかし、トランザクション・サーバ４６８に送信されたアカウンティング情報は、様々な理由から、不完全であり、あるＩＳＰと次のＩＳＰでは異なり、複数回送信される、などの可能性があるということを理解されたい。したがって、同一の単一ユーザ・セッションに関する様々な異なる、また不完全である可能性のある記録が、トランザクション・サーバ４６８によって受信される場合がある。
【０１３７】
当然ながら、特定のユーザ・セッションから発生するすべてのトランザクション・データ記録を識別または関連付けることは、設定システム４７４が請求書を作成し、それらを顧

客４５６に分配し、その結果、顧客４５６が設定システム４７４に支払うことができ、同様に自身の顧客に対しても適宜請求書を作成することができるという点で有利である。同様に、設定システム４７４は、ローミング・ユーザが使用する利益を生むアクセス・タイムに関して遠隔（またはビジター）ＩＳＰまたは他のサービス・プロバイダ４５２に対して支払いを行う。設定システム４７４は、ローミング・ユーザが認可した使用に対する支払いもさらに保証することができる。したがって設定システム４７６の運営者は、複数の関係者間でサービス・アクセス・トランザクションの金銭上の決済を容易にするための機構を提供する安全で信頼性のあるエンティティとして動作する。設定システム４７６は、タイムリーで自動的かつ便利な方法で設定を可能にするために多数の自動機能および動作を実現する。そのような設定またはサービス・アクセス・トランザクションを容易にする設定システムの動作に関するさらなる詳細について以下で説明する。

【０１３８】

物理的アーキテクチャ

図１５は、本発明の一実施形態によるアクセス・ブローカー・システム４５４の物理的アーキテクチャを示す図である。複数のトランザクション・サーバ４６８は、それぞれが関連するデータベース８１２にアクセスする１つまたは複数のサーバ・マシン８１０に常駐しているように示されている。サーバ・マシン８０６に常駐しているウェブ・サーバと電話帳サーバは、遠隔内部ユーザ８１４および顧客４５６によってアクセス可能である。ウェブ・サーバは、ウェブ・ページ（例えばＨＴＭＬ文書）を生成し、遠隔内部ユーザ８１４と顧客４５６の両方に配信するように動作する。上記のように、本発明の一実施形態では、マシン８０６に常駐しているウェブ・サーバ上のサーブレットは、一意の接続アプリケーションＩＤをダイヤラーＩＤの一例である形態で、サービス・プロバイダ４５２に常駐している各ダイヤラーまたは接続アプリケーション４６６に提供する。電話帳サーバ（電話帳管理システム４８０の一部）は、顧客４５２の電子電話帳を維持・更新するように動作し、したがってサービス・プロバイダ４５２への／からの更新を受信／発行し、そのような更新を顧客４５６に発行する。

【０１３９】

設定システム４７６および内部ユーザ８１６の集合は、ファイヤーウォール８１８の背後に常駐しているように示されている。具体的には、設定システム４７４は、中央データベース８２２にアクセスする１つまたは複数のサーバ・マシン８２０上のホストとなる。

【０１４０】

概要－設定システム

図１６は、本発明の一実施形態による設定システム４７４のアーキテクチャを示すブロック図である。設定システム４７４は、バックエンド・アプリケーション８２４、フロントエンド・アプリケーション８２６、データ蓄積／報告アプリケーション８２８、およびシステム・インターフェース８３０を含んでいる。

【０１４１】

バックエンド（またはサーバ側）・アプリケーション８２４は、トランザクションの価格を設定し、１つのトランザクションに関与するすべての関係者の残高を更新し、信用限度を検証する設定アプリケーション８３２と、アカウンティング・サイクルを終了し、定期料金を適用し、インボイスおよび通話詳細記録（ＣＤＲ）を含む請求書報告を生成し、請求書報告をウェブに発行する請求書作成アプリケーション８３４と、ビジネス・ルールと中央データベース８２２の構造上の保全性とを検証する監査アプリケーション８３６とを含むように示されている。設定アプリケーション８３２は、柔軟な価格設定エンジン４７６を実施するように示されている。

【０１４２】

本実施形態では、設定アプリケーション８３２は、正規化機能、集計機能、および検証機能を担当する。正規化機能には、複数のトランザクション・サーバ４６８から受信したアカウンティング・データを請求書作成のために使用される単一形式ＣＤＲに変換すること、サービス・アクセス・トランザクションに関与する関係者を識別すること、および特定

のサービス・アクセス・トランザクションに関してアクセス・ブローカー・システム４５４がプロバイダ４５２に対して支払い義務のある価格と顧客４５６がアクセス・ブローカー・システム４５４に対して支払い義務のある価格とを定義することが含まれる。集計機能には、サービス・アクセス・トランザクションに関与するすべての関係者について購入価格と販売価格を残高に記入すること、および適切な残高を更新することを必要とする。検証機能には、信用限度の検証が含まれる。

【０１４３】

設定システム４７４は、プロバイダ４５２と顧客４５６の両方による略リアルタイムの収入／勘定追跡を可能にするサービス・アクセス・トランザクションの略リアルタイムの設定を提供するよう動作する。

【０１４４】

本発明のある種の実施形態では、設定システム４７４は、以下の特徴を有する柔軟な価格設定モデルをサポートする柔軟な価格設定エンジン４７６を含む。

１．顧客４５６、サービス・プロバイダ４５２、サービス・アクセスの位置、サービス・アクセスの種類（例えばダイヤルアップ・モデム、ＩＳＤＮ、ＤＳＬ）、または特定の顧客４５６について特定周期中に蓄積された使用量などによって異なる多彩なデータ構造。

２．（ａ）使用量による（例えば速度およびセッションの長さに応じた）価格設定、（ｂ）トランザクションによる（トランザクションあたりいくらという）価格設定、および（ｃ）加入ベースの、すなわち均一な価格設定（例えば顧客４５６に対する１回の請求書作成周期中の全使用量に対する１つの価格、または請求書作成周期中の顧客に対するユーザごとの１つまたは複数の価格）。

３．申し込まれた割引および販売促進。

４．開業料金、月次料金、および最低月次手数料のような様々な料金。

５．マルチティアの価格設定方式、または内部プロバイダ・ローミング。ここで特定の位置に関する買い相場と売り相場はプロバイダ４５２によって異なり、またサービス・アクセスのサービス・ユーザ／顧客４５６がさらに別の顧客４５６、その提携者、またはそれらの顧客に属しているか否かによって異なる。

【０１４５】

この柔軟な価格設定エンジン４７６はデータベース主導によるものであり、したがって適切な計画を中央データベース８２２内で維持されている価格設定テーブル（図示せず）にロードすることによって新しい価格設定モデルの実施を可能にしている。より具体的には、柔軟な価格設定エンジン４７６は、マルチティアの価格設定モデルに役立ち、それによって単一サービス・アクセス・トランザクションに対する相場は、複数の基準によって消費者（または顧客）の複数のティア(tier)全体に適用することができる。これらの基準には、特に、使用量（例えば蓄積された使用時間または値の合計）による価格設定、およびトランザクション（例えば蓄積されたトランザクションの合計数）による価格設定の任意の組み合わせを含めることができる。

【０１４６】

図１６およびフロントエンド・アプリケーション８２６に戻ると、データ管理アプリケーション８３８は、ビジネス・プロセスを実行し、情報目的でデータを閲覧するためにアクセス・ブローカーの様々な機能ユニットによって使用される。この目的で、データ管理アプリケーション８３８は、顧客４５６およびアクセス・ポイントに関する情報を管理し、アカウンティング機能および経営管理機能を実行するために様々なユーザ・インターフェースを提供することができる。

【０１４７】

注文処理アプリケーション８４０は、新しい法人顧客に注文を出すために顧客４５６（例えばソリューション・パートナー４８８または再販業者）にユーザ・インターフェースを提供する。

【０１４８】

データ蓄積／報告アプリケーション８２８は、運営報告、機能報告、およびネットワーク

・ロード報告を可能にするように日毎、または月毎にデータを集計する複数の処理を含む。

【０１４９】

システム・インターフェース８３０は、トランザクション・サーバ・ローダー８４２、プロバイダ・ローダー８４４、およびアカウンティング・システム・インターフェース（図示せず）を含むローダー・アプリケーションを有する。まずトランザクション・サーバ・ローダー８４２を扱うことによって、「データ・ローダー」構成要素は、それぞれのトランザクション・サーバ４６８のデータベース８１２から中央データベース８２２に、一意のセッションＩＤを含めてトランザクション・データ記録の形式でアカウンティング記録を処理するためにプルする。複数のトランザクション・サーバまたはバッチ・ローダー８４２は、分散型データベース・リンクとして実施することができ、アカウンティングまたはトランザクション・データ記録は略リアルタイムでローダー８４２を介してプルされる。

【０１５０】

概要－データ・モデル

図１７Ａは、顧客テーブル８４６、アクセス・ポイント・テーブル８４８、価格設定テーブル８５０、ＣＤＲテーブル８５２、アカウンティング・テーブル８５４、認証トランザクション記憶領域またはテーブル８５６、バッチ履歴記憶領域またはテーブル８５８、およびＳＱＭ記憶領域またはテーブル８６０を含むデータ・モデル例８４５を示すブロック図である。

【０１５１】

アクセス・ブローカー・システム４５４のネットワーク構成要素は、ある種の実施形態では、トランザクション・データ記録から経路指定プレフィックスを取り除くことができる。これらの構成要素のいくつかは、ユーザＩＤストリングの末尾を切り捨てることもできる。一意のセッションｉｄプレフィックスは、経路指定プレフィックスではなく、またユーザ名の末尾にあるわけでもないので、取り除かれることも末尾が切り捨てられることもない。したがってユーザＩＤストリングは、それがユーザ・セッションを一意に定義するよう使用される前にこれらの欠点を除去するために処理される。変更されたユーザ・セッションＩＤは、使用可能な以下の構成要素の多くを使用して構築される。

＜ＡｕｔｈＣｕｓｔｏｍｅｒＩｄ＞／＜ＵｎｉｑｕｅＩＤ＞／［ＣｕｓｔｏｍｅｒＰｒｅｆｉｘ（ｓ）］／＜ＥｎｄＵｓｅｒＮａｍｅ＞@＜ＮｏｎＲｏｕｔｉｎｇＣｕｓｔｏｍｅｒ　Ｄｏｍａｉｎ＞

ここで、

＜ＡｕｔｈＣｕｓｔｏｍｅｒＩｄ＞は、顧客ソリューション・プロセスによって作られた認証顧客ＩＤである。

＜ＵｎｉｑｕｅＩＤ＞は、上記のような接続アプリケーション４６６によって生成された一意のセッションＩＤコード、０Ｕｘｘｘｘｘｘｘｘ／、プレフィックスである。

＜ＣｕｓｔｏｍｅｒＰｒｅｆｉｘ（ｓ）＞は、上記のような内部経路指定のために顧客が使用するプレフィックスである。

＜ＥｎｄＵｓｅｒＮａｍｅ＞は、上記のようなアクセス・ブローカー・システム４５４に接続しているエンド・ユーザのユーザＩＤである。

＜ＮｏｎＲｏｕｔｉｎｇＣｕｓｔｏｍｅｒＤｏｍａｉｎ＞は、上記のような内部経路指定のために顧客が使用するドメインである。

【０１５２】

特に図１７Ｂを参照すると、プロバイダ・ローダー８４４は、一意のセッションＩＤを含む通話詳細記録（ＣＤＲ）またはトランザクション・データ記録をプロバイダ４５２からバッチ形式で受信する。ＣＤＲデータはプロバイダ・ローダー８４４によって処理される。プロバイダ・ローダー８４４は適切なＦＴＰサイトからデータを取り出し、それをトランザクション・サーバ４６８から受信したデータと同じ形式に変換することができる。具体的には、トランザクション・サーバ４６８は、ユーザ・アクセス・セッションが上記の

ように認証されるたびに変更されたユーザ・セッションＩＤを構築し、それを認証トランザクション・テーブル８５６のｓｅｓｓｉｏｎ＿ｉｄフィールドに、またアカウント・トランザクション・テーブル８５４のｓｅｓｓｉｏｎ＿ｉｄフィールドに記憶する（図１７Ａを参照のこと）。認証トランザクション・テーブル８５６に記憶されている変更されたセッションＩＤごとに、対応するトランザクション・データ記録は、処理するために設定システム４７４によって受信されるべきであるということが理解されよう。トランザクション・サーバ４６８と類似の仕方で、バッチ・ローダー８４２、８４４はそれぞれに、サービス・プロバイダ４５２のトランザクション・サーバ４６８から受信した各トランザクション・データ記録から変更されたトランザクション・データ記録を組み立て、または構築する（図５および１７Ｂを参照のこと）。ローダー８４２、８４４からの変更されたセッションＩＤは、バッチ履歴テーブルのｓｅｓｓｉｏｎ＿ｉｄフィールドに記憶される。同様にして、ＳＱＭプロセスは、変更されたセッションＩＤを構築し、それをＳＱＭテーブル８６０のｓｅｓｓｉｏｎ＿ｉｄフィールドに記憶する。

【０１５３】

一意のコードを含めて一意のセッションＩＤの使用は、以下の問題に対処する際に使用することができる。

【０１５４】

欠落したアカウンティング記録

欠落したアカウンティング／トランザクション・データ記録（図１９のブロック８６２を参照のこと）は、配信の失敗、変形した記録、誤った経路指定など様々な理由で発生する可能性がある。配信の失敗は、ＩＳＰからのインターネット接続（例えば図６のネットサーバ４７０）が中断され、したがって設定システム４７６へのトランザクション・データ記録の配信がブロックされた場合に発生する可能性がある。数分以上の長期にわたって続く接続の停止は、通常、最低限の送信再試行機能が原因でネットサーバ４７６にトランザクション・データ記録を破棄させることになる。ＲＡＤＩＵＳプロトコルを使用する際、変形した記録は、通常、サービス・プロバイダの認証サーバを含めて複数の中間点のどこでも破棄される（例えば認証／認可サーバ６０２またはネットサーバ４７０（図６を参照のこと））。誤って経路指定された記録は、偶然または不正な目的でＩＳＰ認証サーバ６０２の構成が不適切であるために送信されない記録である。

【０１５５】

ユーザによるすべてのアクセス・セッションは最初に認証を必要とするので、認証トランザクション・テーブル８５６の各ｓｅｓｓｉｏｎ＿ｉｄフィールドはａｃｃｔ＿ｔｒａｎｓテーブルに対応するｓｅｓｓｉｏｎ＿ｉｄフィールドを含むべきである。したがって、ｓｅｓｓｉｏｎ＿ｉｄフィールドを関連付け、照合し、相関関係を証明し、精査することにより、欠落したアカウンティング／トランザクション・データ記録を決定することができる。図示する実施形態では、欠落したアカウンティング／トランザクション・データ記録は、通常、認証トランザクションすなわちａｕｔｈ＿ｔｒａｎｓテーブル８５６に認証要求記録を有するが、アカウンティングすなわちａｃｃｔ＿ｔｒａｎｓテーブル８５４には一致するアカウンティング開始および／またはアカウンティング停止記録は有しない。したがって、ａｕｔｈ＿ｔｒａｎｓテーブルの各ｓｅｓｓｉｏｎ＿ｉｄフィールドに対応するａｃｃｔ＿ｔｒａｎｓテーブルのすべてのｓｅｓｓｉｏｎ＿ｉｄフィールドを検索することによって、欠落したアカウンティング／トランザクション・データ記録を発見することができる（ブロック８６４～８７２を参照のこと）。

【０１５６】

不適切なアカウンティング記録

不適切なアカウンティング／トランザクション・データ記録は、一般に図６のサーバ６００などのプロバイダの認証サーバ（ＡＡＡサーバ）の不適切な構成が原因で設定システム４７４（図６を参照のこと）によって受信される場合がある。不適切な構成により、通常、プロバイダの認証サーバ６００は、すべてのアカウンティング／トランザクション・データ記録をユーザ・アクセス・セッションの認証を担当するプロキシだけにではなく、す

べてのプロキシに送信することになる。この場合、不正確な受信者のａｕｔｈ＿ｔｒａｎｓテーブルにはｓｅｓｓｉｏｎ＿ｉｄフィールドがまったくない。これらのアカウンティング／トランザクション・データ記録は、通常、対応する認証記録を有していないので、これらは比較的容易に識別することができ、また例えば顧客サポート・チームは、プロバイダの構成上の問題を解決することができ、そのような不正確な伝送の再発を防止することができる。これらの場合、図１９に示す方法を使用することができる。ただし、ａｃｃｔ＿ｔｒａｎｓテーブルのエントリに対応する一意のセッションＩＤを見つけるためにａｕｔｈ＿ｔｒａｎｓテーブルが検索される。

【０１５７】

重複したアカウンティング記録

重複したアカウンティング記録は、同一の単一アクセス・セッションを記述する複数のトランザクション・データ記録である。図示する実施形態では、重複したトランザクション・データ記録は、各リアルタイムのアカウンティング／トランザクション・データ記録の６個の「鍵」フィールドを過去３０日以内に受信したすべての他のリアルタイム・アカウンティング／トランザクション・データ記録と照合する比較的単純なアルゴリズムを使用して設定システム４７４によってアクティブにフィルタリングされる。使用されるフィールドの例としては、ＲＡＤＩＵＳセッションＩＤ、プロバイダＩＤ、ＮＡＳ　ＩＰアドレス、ユーザ、ドメイン（ユーザａｕｔｈ領域）およびセッション・タイムがある。

【０１５８】

ある種の実施形態では、すべての６個のフィールドがＣＤＲテーブルの格付け済み記録内のフィールドと一致した場合、現行の記録は重複とマーク付けされ、破棄される。

【０１５９】

重複したアカウンティング記録は、以下の様々な理由から生じる可能性がある。

【０１６０】

アカウンティング／トランザクション・データ記録は、送信者に対してアカウンティング－応答メッセージを適宜送信することによって確認応答する必要がある。残念ながら、「適宜」はＲＡＤＩＵＳの仕様によっては定義されず、数秒後、数時間後、または数日後に異なるベンダーおよび構成が確認応答していないアカウンティング・トランザクションを再送する場合がある。アカウンティング要求は設定システムによって実際に受信されたが、確認応答が消失または変形した場合、発信者はアカウンティング記録の複数のコピーを再送する場合がある。そのような記録はすべて、受信側トランザクション・サーバ４６８によって補足され、最終的には処理するために設定システム４７４によって取り出される。このクラス特有の変形が発生する場合があるが、これは設定重複フィルタリング・アルゴリズムを逃れるものであり、送信側ＮＡＳ５３２は再送のたびに更新された（例えば増分されて長くなった）セッション・タイムを送信するか、またはＲＡＤＩＵＳセッションＩｄが再送と再送の間で変更される。場合により、ＮＡＳ５３２は一定したＮＡＳ　ＩＰアドレスを送信せず、その場合、アクセス・セッションをサービス・プロバイダすなわちＩＳＰに関連付けるために別の属性（例えばコールド・セッションＩｄまたはプロバイダＩｄ）が使用される。このような場合、重複した検出のためにＮＡＳ　ＩＰの有用性が低減される。

【０１６１】

重複したアカウンティング記録は「バッチ」プロバイダによって送信することができるが、この「バッチ」プロバイダのアカウンティング供給は重複のないものと仮定される。上記理由の１つにより、リアルタイム・アカウンティング・プロバイダがアカウンティング記録を配信することに失敗した場合、そのアカウンティングの責任を全うするためにそのリアルタイム・アカウンティング・プロバイダによって記録のバッチが送信された場合、重複したアカウンティング記録を設定システム４７４に手動で挿入することができる。これらの場合、サービス・プロバイダ４５２は任意のデータセットを送信することができるが、この任意のデータセットはデータ正規化プロセスへの提出に備えてアクセス・ブローカー・システム４５４の要員が特別に処理する必要がある。このようなデータセットは、

以前に報告されたセッションと、訂正が試行されている欠落したセッションの両方を記述するデータを含むことができる。これらの記録バッチは通常は単発で事前処理されるので、重複した挿入を防止するための管理はほとんどない。このプロセスは、一意のセッションＩＤに鑑みて自動化することができるということが理解されよう。

【０１６２】

いくつかのサービス・プロバイダは、鍵重複フィールドを不規則に使用することを原因に重複したトランザクション・データ記録をアクセス・ブローカー・システム４５４に入れることを許可する場合がある。例えばある種の状況では、サービス・プロバイダはＮＡＳ　ＩＰ属性をランダムなデータで充填し、したがって重複フィルタリング基準が悪影響を受ける。矛盾したセッションｉｄの生成またはユーザ切断時におけるセッション期間の調整失敗のような他の変則事項により、別個のセッションに対応するように見える重複が生成される場合がある。ここでもまた、一意のセッションＩＤはこれらの問題を解決する際に役立つ場合がある。

【０１６３】

本明細書で例示するように、重複アカウンティング／トランザクション記録は、各リアルタイム・アカウンティング／トランザクション・データ記録の６個の「鍵」フィールドを過去３０日以内に受信されたすべての他のリアルタイム・アカウンティング／トランザクション・データ記録と照合するアルゴリズムを使用して設定システム４７４によってアクティブにフィルタリングされる。それぞれの承認された単一セッションを一意に識別する一意のｓｅｓｓｉｏｎ＿ｉｄフィールドを使用することによって、精度を高めることができる。

【０１６４】

重複エイリアス記録

重複を検出するアルゴリズムがある記録を重複であると不適切に識別した場合、重複エイリアス記録が発生する。例えばサービス・プロバイダのＮＡＳ（例えば図６のＩＳＰ　５１０のＮＡＳ５３２）が短期間にセッションＩＤデータ値を生成または再利用しない場合、このようなケースが生じる可能性がある。信頼性のないサービス・プロバイダによって生成されたいかなるセッションＩＤに加えて、またはこれの代わりに、各単一アクセス・セッションを一意に定義する本発明の実施形態の一意のセッションＩＤは、重複したエイリアス記録の発生を低減させるために重複検出アルゴリズムによって使用することができる。具体的には、各セッションが一意に識別される際に重複エイリアス記録を少なくとも実質的に低減するためにｓｅｓｓｉｏｎ＿ｉｄテーブル内の変更された一意のセッションＩＤを使用することができる。

【０１６５】

無効セッション－長さ記録

アクセス・セッションの期間に関する設定システム４７４によって受信されたすべてのアカウンティング／トランザクション・データ記録は、いつも完全であるとは限らないということが理解されよう。例えばアカウンティング／トランザクション・データ記録は、セッション期間データ（例えばＡｃｃｔ－Ｓｅｓｓｉｏｎ－Ｔｉｍｅ属性）が欠落しており、０値を含んでおり、セッションに対する不正確な値（例えばあるセッションは実際には３分の長さなのに４分の長さであると報告すること）を含んでおり、ＲＦＣ　２１３９、セクション５．７で規定されるような過度に大きな値を含んでいるか、または無効であるなどの場合がある。無効アクセス期間は、例えばサービス・プロバイダのモデム・バンクがユーザによる切断を報告せず、別のセッションが同じ物理的なモデム・ポートで開始されるか、または何らかの他の理由からタイムアウトが発生するまでＮＡＳ５３２がセッション・タイムを蓄積し続ける場合に発生する可能性がある。

【０１６６】

重複した無効セッションの長さを有するアカウンティング／トランザクション・データ記録は、例えば以下のような様々な理由から生じる場合がある。

【０１６７】

欠落したＡｃｃｔ－Ｓｅｓｓｉｏｎ－Ｔｉｍｅ
アカウンティング／トランザクション・データ記録がネットサーバ４７０によって受信され、Ａｃｃｔ－Ｓｅｓｓｉｏｎ－Ｔｉｍｅ属性が欠落している場合、ネットサーバ４７０は、通常、０セッション長を有するアカウンティング／トランザクション・データ記録を転送する。
【０１６８】
不正確なＡｃｃｔ－Ｓｅｓｓｉｏｎ－Ｔｉｍｅ
ＮＡＳ５３２による不正確な時間のアカウンティング、またはサービス・プロバイダによる意図的な不正行為によって、不正確なセッション期間を有するアカウンティング／トランザクション・データ記録が生成される場合がある。
【０１６９】
Ａｃｃｔ－Ｓｅｓｓｉｏｎ－Ｔｉｍｅ０
０値のＳｅｓｓｉｏｎ－Ｔｉｍｅ属性を有するアカウンティング／トランザクション・データ記録がネットサーバ４７０によって受信されると、ネットサーバ４７０は、通常、０セッション長を有するアカウンティング／トランザクション・データ記録を転送する。
【０１７０】
長いＡｃｃｔ－Ｓｅｓｓｉｏｎ－Ｔｉｍｅ
不正行為、誤作動、または不適切な構成が原因でセッション・タイム・アカウンティングは法外な長さのセッションを識別する場合がある。
【０１７１】
切断検出の失敗
「長い」セッションまたは同一期間を有する複数のセッションは、サービス・プロバイダのモデム・バンクの誤動作および／またはユーザが長期間切断されていることを検出することができないその不適切な構成が原因で発生する場合がよくある。
【０１７２】
不正アクセス
長期のセッション・タイムは、悪意のあるユーザによる連続した使用が原因で発生する場合もある。
【０１７３】
破損したＡｃｃｔ－Ｓｅｓｓｉｏｎ－Ｔｉｍｅ
ＮＡＳ５３２、サービス・プロバイダの認可／認証サーバ６００、またはサービス・プロバイダのネットサーバ４７０によるフィールド処理でのエラーは、アカウンティング／トランザクション・データ記録のセッション・タイム属性を破損する可能性がある。実際のサンプルに基づき、何らかの先行ＲＡＤＩＵＳパケットに長い、ベンダー特有のデータがある場合に発生することが多い。
【０１７４】
本物の長いセッション
長いセッションのフィルタリングの精度は、フィルター閾値（通常は約１００時間）によって異なる。
【０１７５】
ＳＱＭ記録に提供されたセッション長をアカウンティング記録に提供されたセッション長に対して相関関係を証明し、関連付けるなどによって精度を強化することができる。各トランザクション・データ記録はその一意のセッションＩＤを有しているので、セッション長はａｃｃｔ＿ｔｒｎｓテーブルのｓｅｓｓｉｏｎ＿ｉｄフィールドとＳＱＭテーブル８６０のｓｅｓｓｉｏｎ＿ｉｄフィールドを使用して関連付けられた記録から取得することができる。したがって、あるトランザクション・データ記録で欠落しているデータは、一意のセッションＩＤを有する別のトランザクション・データ記録から取得することができる。
【０１７６】
アカウンティング記録のオーバーラッピング

ある場合には、トランザクション・データ記録は、時間的にオーバーラップしている同じユーザの信用証明書（例えば同じユーザ名およびパスワード）を含むサービス・プロバイダから受信される。本発明の本実施形態では、各アクセス・セッションは一意のセッションＩＤを含むので、オーバーラップしているトランザクション・データ記録の分析が容易になる場合がある。具体的には、それらの記録のセッションの詳細を決定するためにａｃｃｔ＿ｔｒａｎｓテーブル８５４のｓｅｓｓｉｏｎ＿ｉｄフィールドとＳＱＭテーブル８６０のｓｅｓｓｉｏｎ＿ｉｄフィールドを使用することができる。例えば、一意のセッションＩＤを使用することによって、そのようなセッションが２つの本物の別個のセッションであるか否か、あるいはそのセッションが異常なＮＡＳ５３２が原因で生成されたものか否かを判定することができる。

【０１７７】

真偽が問われる記録

セッションＩＤは各単一のセッションを一意に識別し、一意のセッションＩＤを含むセッションに関するデータはシステム内の様々な異なるサーバに送信されるので、真偽問題の解決が容易になる場合がある。具体的には、顧客サポート・チームは、認証または認可されたトランザクション、アカウンティングおよびＳＱＭテーブル８５６、８５４、および８６０のセッション詳細をそれぞれに比較することができ、これによって単一ユーザ・アクセス・セッションに一意に関連付けられたトランザクション・データの３つの異なるソースが提供される。これらテーブルのｓｅｓｓｉｏｎ＿ｉｄフィールドは、特定ユーザのアクセス・セッションの詳細を裏付けるために関連付けまたは相関関係の証明などに役立つ。

【０１７８】

チャレンジ・プロバイダ記録の記録品質

各セッションは一意に識別され、様々な異なるサーバはトランザクション・データ記録を設定システム４７４に独立して通信するので、特定のサービス・プロバイダから受信したトランザクション・データ記録の品質は、その特定のサービス・プロバイダからのトランザクション・データ記録を他のソースからの記録と比較することによって評価することができる。これは、そのようなアカウンティング機能に関する問題、またはサービス・プロバイダの技術問題に関する問題を分離する際にネットワーク・アクセス・チームに役立つ場合がある。

【０１７９】

複数人員による合法的Ｉｄの使用

一意のセッションＩＤが各単一のユーザ・セッションを一意に識別するので、単一ユーザＩＤデータの合法的使用法が複数のユーザによって使用される別個のユーザ・アクセスの発生を識別するために使用することができる。したがって、４５６顧客は、例えば組織が小規模であり、かつ／またはそのような仕方で動作することを選択するという理由で、同一のログイン名を共有する場合がある。このような場合、開始時が同時でセッション長が同じ複数の位置からのログインがある場合がある。一意のセッションＩＤの内包は、それらの状態を高度な精度をもって精査するために使用される。

【０１８０】

ダイヤラーのポリシー管理バージョニング

ＳＱＭ記録をアカウンティング記録に関連付け、相関関係を証明するなどのために一意のセッションＩＤを使用することができる。これにより、ＳＱＭ記録なしに作成されたアカウンティング記録を、アクセス・ブローカー・システム４５４によって関連付けられない、あるいは提供されない、接続アプリケーション（例えば接続アプリケーション４６６）の使用、またはそのサポートされていないバージョンの使用を明らかにするために使用することができる。通常、接続アプリケーション４６６のサポートされていないバージョン（例えば接続ダイヤラー技術）を使用している顧客および個別ユーザに関して報告が作成される。これは、そのようなユーザをより最新のバージョンに移動させるために役立つ。このような状況では、顧客４５６が不適切な接続アプリケーション４６６を使用している

場合、その顧客に関連付けられたアカウントを自動的にディセーブルすることができる。したがって、その問題を識別するために、アクセス・ブローカー・システム４５４に連絡を取り、その結果バージョンの移動を強制することを顧客４５６に強制することができる。

【０１８１】

全体的な請求書作成プロセスの品質の改善

したがって、単一ユーザ・セッションに関連付けられたすべてのトランザクション・データ記録を一意に識別する一意のセッションＩＤを包含することによって、トランザクション記録の処理の精度が高められるということが理解されよう。したがって、請求書作成の真偽の問題が生じることは少なくなり、発生する可能性のあるいかなる真偽の問題の解決法もより迅速に確定することができる。

【０１８２】

図１３は、２０５、３０５、または４０５のようなネットワーク・アクセス・デバイス、ネットワーク・ダイヤラー４６６、または２４０、３５０、４６８、または４７２のようなネットサーバとして構成することができるコンピュータ・システム７００を示す図である。コンピュータ・システム７００は、システム・バス７４５を介してランダム・アクセス・メモリ（ＲＡＭ）７３５および読み出し専用メモリ（ＲＯＭ）７４０に動作可能に接続されているプロセッサ７５０を含む。プロセッサ７５０は、入出力バス７３０を介して、光ディスクまたは他の記憶媒体であってよい固定ディスク７２０にも接続されている。あるいは、プロセッサ７５０は、入出力バス７３０を介して複数の記憶デバイスに接続することができる。プロセッサ７５０は、システム・バス７４５および入出力バス７３０を使用してデータを通信する。

【０１８３】

システム・バス７４５と入出力バス７３０は、キーパッド７２５または入力デバイス７１０からの入力を受信することもできる。システム・バス７４５と入出力バス７３０は、ディスプレイ７０５、固定ディスク７２０、および／または出力デバイス７１５に出力を提供することができる。メモリおよび記憶媒体７３５、７４０は、フラッシュ・メモリ、ＥＥＰＲＯＭ、または上記のいかなる組み合わせも含むことができる。

【０１８４】

コンピュータ・システム７００は、オペレーティング・システム・ソフトウェアの一部であるディスク・オペレーティング・システムのようなファイル管理システムを含むオペレーティング・システム・ソフトウェアによって制御することができる。ファイル管理システムは、ＲＯＭ７４０のような不揮発性記憶デバイスに記憶することができ、データを入力および出力し、そのデータをＲＡＭ７３５およびＲＯＭ７４０上に記憶するためにオペレーティング・システムによって要求される様々な機能をプロセッサ７５０に実行させるよう構成することができる。コンピュータ・システム７００の一実施形態では、プロセッサ７５０に、ネットワーク・アクセス・デバイス２０５または３０５のようなネットワーク・アクセス・デバイスの機能を実行させる命令を固定ディスク７２０またはＲＯＭ７４０に記憶することができる。代替形態では、プロセッサ７５０に、ネットサーバ２４０、３５０、４７２、または４６８のようなネットワーク暗号解読サーバの機能を実行させる命令を固定ディスク７２０またはＲＯＭ７４０に記憶することができる。

【０１８５】

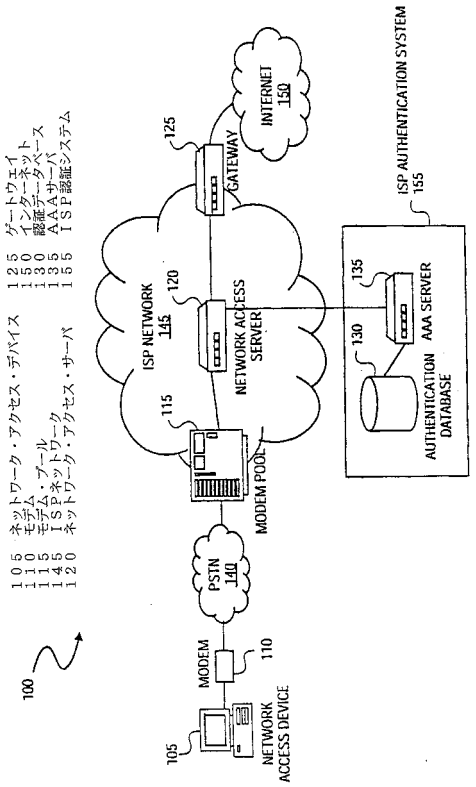
以上、サービス・アクセス・システムで生成された複数のトランザクション・データ記録を関連付ける方法およびシステムを開示した。上記の詳細な説明では、本発明はその特定の実施形態を参照して説明した。しかし、首記の特許請求の範囲に示す本発明のより広範な範囲および趣旨を逸脱せずに、本発明に様々な修正形態および変更を行うことができることが明らかになろう。したがって本明細書および図面は、限定の意味ではなく説明の意味でみなされたい。

【図面の簡単な説明】

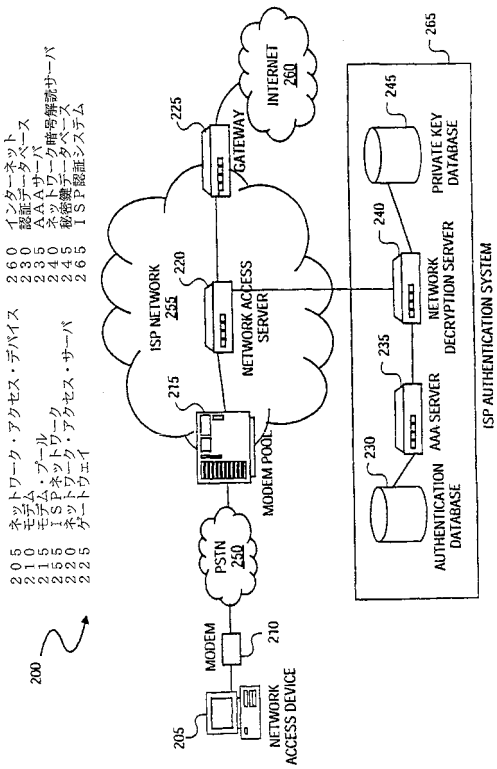
【０１８６】

【図１】安全でない方法を使用してネットワーク・ユーザの信用証明書が認証される従来技術のＩＳＰネットワーク構成を示す図である。
【図２】本発明の一実施形態と調和する、ＩＳＰネットワーク、ネットワーク・アクセス・デバイス、およびネットワーク暗号解読サーバ含むネットワーク構成を示す図である。
【図３】本発明の一実施形態と調和する、遠隔ＩＳＰネットワーク、ネットワーク・アクセス・デバイス、およびネットワーク暗号解読サーバを含むネットワーク構成を示す図である。
【図４】ネットワーク・ユーザの信用証明書を安全に認証する方法の一例としての動作の流れ図である。
【図５】多数のサービス・プロバイダ、アクセス・ブローカー・システム、および複数の顧客を含む、本発明の一実施形態によるマルチパーティ・サービス・アクセス環境を示すブロック図である。
【図６】ローミング・インターネット・アクセスを提供するための、本発明の一実施形態によるアクセス・ブローカー・システムの動作を示す概略図である。
【図７】本発明の一実施形態による接続ダイヤラーの概略的な機能ブロック図である。
【図８】暗号解読機能を含む、本発明の一実施形態によるトランザクション・サーバの概略的な機能ブロック図である。
【図９】暗号解読機能を含む、本発明の別の一実施形態による顧客またはローム・サーバの概略的な機能ブロック図である。
【図１０】接続ダイヤラーによって実行される暗号化メソッドの一例の概略的な流れ図である。
【図１１】トランザクション・サーバまたは顧客サーバによって実行される暗号解読方法の一例の概略的な流れ図である。
【図１２】チェックサム・データの暗号化メソッドの一例の概略的な流れ図である。
【図１３】ネットワーク・アクセス・デバイスまたはネットワーク暗号解読サーバとして構成することができるコンピュータ・システムの概略図である。
【図１４】本発明の一実施形態による、ローミング・インターネット・アクセスを提供するアクセス・ブローカー・システムの動作を示す概略的なブロック図である。
【図１５】図１４のアクセス・ブローカー・システムの物理的アーキテクチャ例の概略的なブロック図である。
【図１６】設定システム例の概略的なブロック図である。
【図１７Ａ】アクセス・ブローカー・システムで使用されるデータ・モデル例を示す図である。
【図１７Ｂ】本発明による一意のセッションＩＤを使用した、同様に本発明による処理を示す概略図である。
【図１８】本発明の一実施形態による一意のセッションＩＤ例を示す図である。
【図１９】一意のセッションＩＤを使用して欠損しているトランザクション・データ記録を識別する方法の概略的な流れ図である。
【図２０】同様に本発明の一実施形態による接続アプリケーションでの一意のセッション識別方法の概略的な流れ図である。

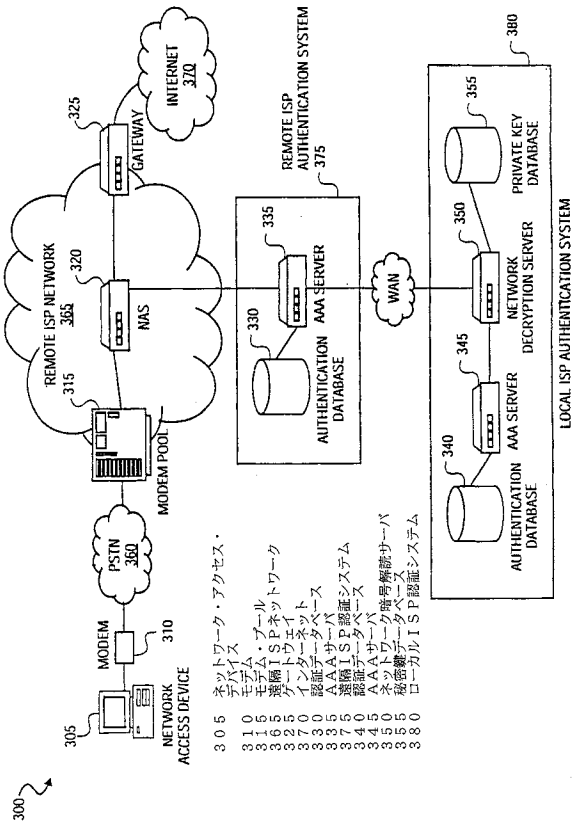
【図１】



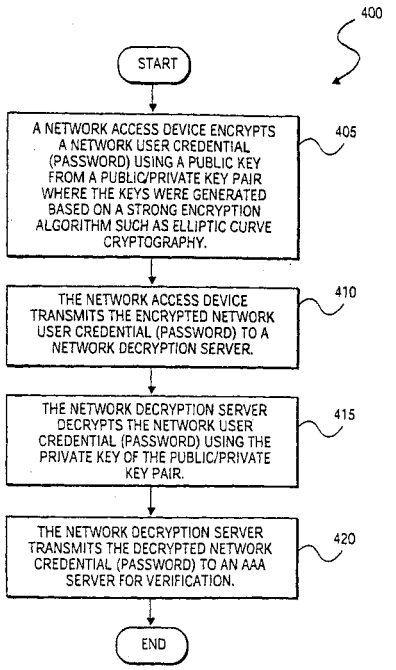
【図２】



【図３】



【図４】



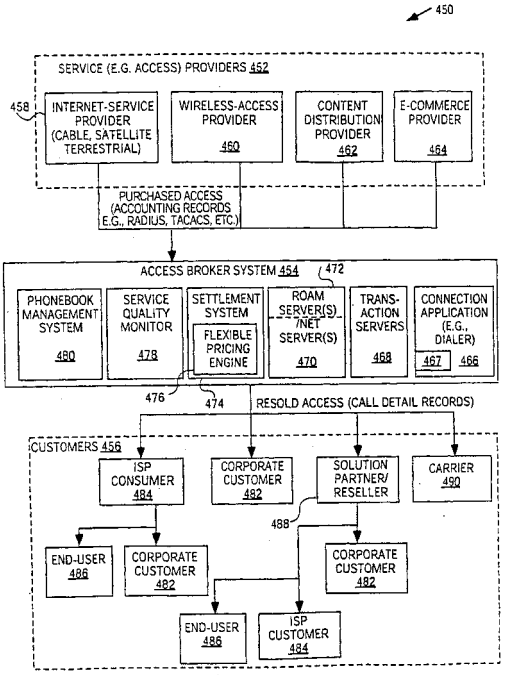
４０５　ネットワーク・アクセス・デバイスは、公開／秘密鍵対の公開鍵を使用してネットワーク・ユーザの信用証明書（パスワード）を暗号化する。ここで、鍵は、楕円曲線暗号法のような強力な暗号化アルゴリズムに基づいて生成されている。

４１０　ネットワーク・アクセス・デバイスは暗号化されたネットワーク・ユーザの信用証明書（パスワード）をネットワーク暗号解読サーバに送信する。

４１５　ネットワーク暗号解読サーバは、公開／秘密鍵対の秘密鍵を使用してネットワーク・ユーザの信用証明書（パスワード）を暗号解読する。

４２０　ネットワーク暗号解読サーバは、暗号解読されたネットワーク信用証明書（パスワード）を検証するためにＡＡＡサーバに送信する。

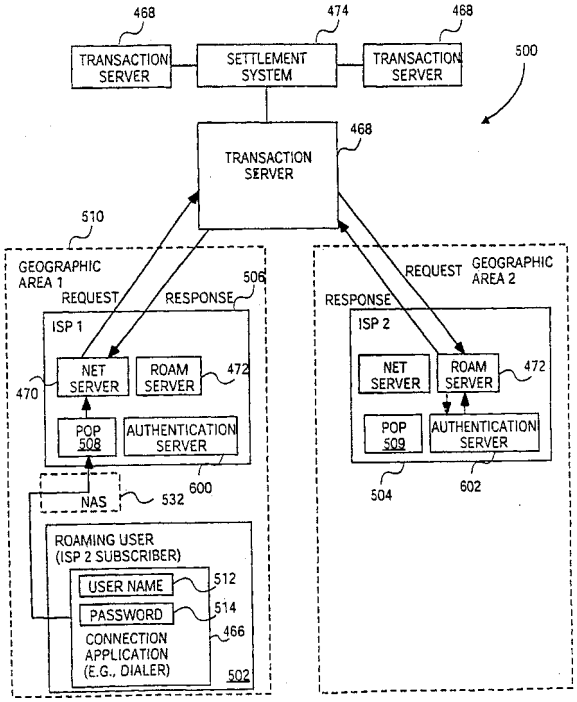
【図５】



452	サービス（例えば、アクセス）プロバイダ		
458	インターネット・サービス・プロバイダ（ケーブル、衛星、地上波）		
460	無線アクセス・プロバイダ	462	コンテンツ配信プロバイダ
464	電子商取引プロバイダ	454	アクセス・ブローカー・システム
480	電話顧客管理システム	478	サービス品質モニタ
474	設定システム	476	柔軟な価格設定エンジン
472	１つ以上のローム・サーバ	470	１つ以上のＮＥＴサーバ
468	トランザクション・サーバ	466	接続アプリケーション（例えば、ダイヤラー）
456	顧客	484	ＩＳＰ消費者
488	ソリューション・パートナー／再販業者	482	法人顧客
486	エンド・ユーザ	490	通信事業者

PURCHARD ACCESS(ACCOUNTING RECORDS E.D., RADUIUS,TACACS,ETC.):
購入されたアクセス（アカウンティング記録、例えば、ＲＡＤＩＵＳ、ＴＡＣＡＣＳ、など）
RESOLD ACCESS(CALL DETAIL RECORDS): 再販されたアクセス（通話詳細記録）

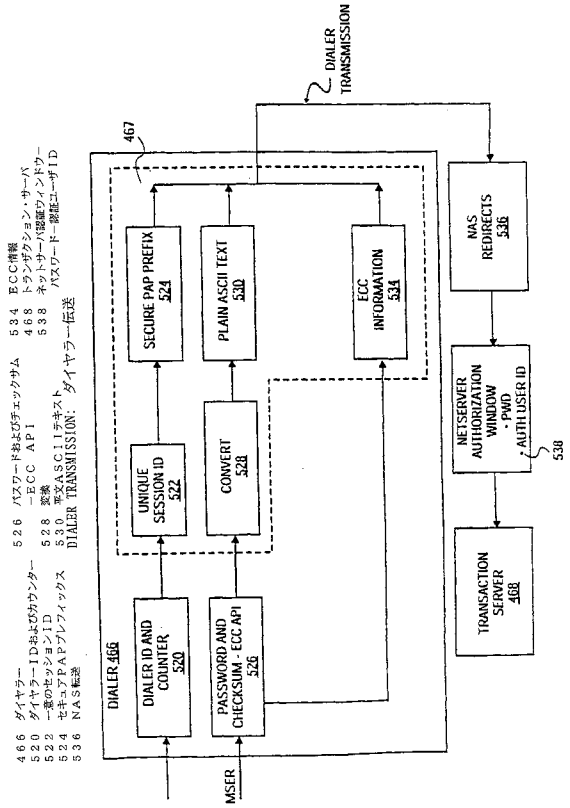
【図６】



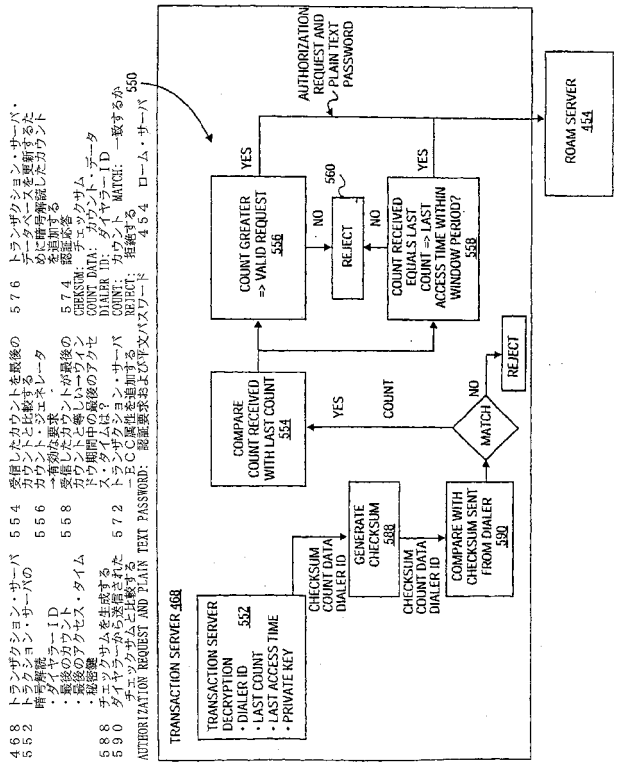
468	トランザクション・サーバ	502	ローミング・ユーザ（ＩＳＰ ２の加入者）
474	設定システム		
510	地理的領域１	512	ユーザ名
470	ネットサーバ	514	パスワード
472	ローム・サーバ	466	接続アプリケーション（例えば、ダイヤラー）
600	認証サーバ		

REQUEST: 要求、RESPONSE: 応答　FEOGRAPHIC AREA: 地理的領域

【図７】

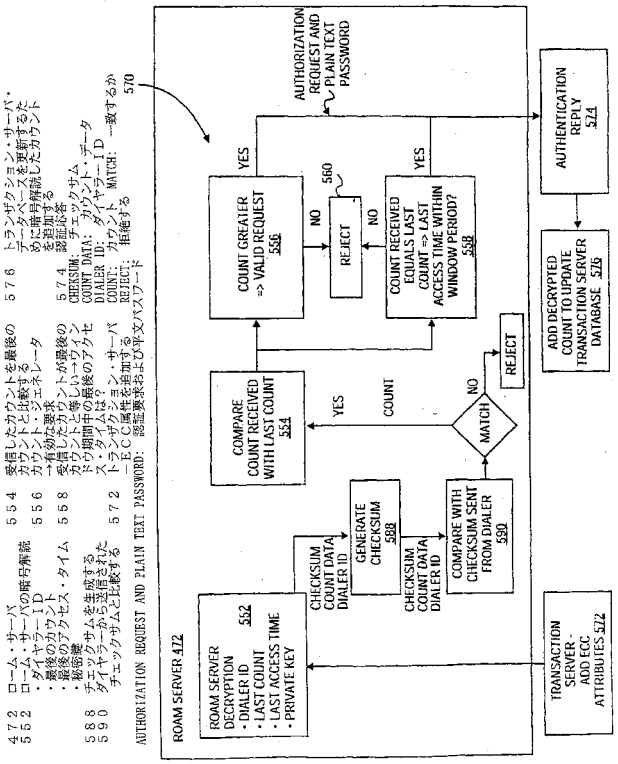


【図８】



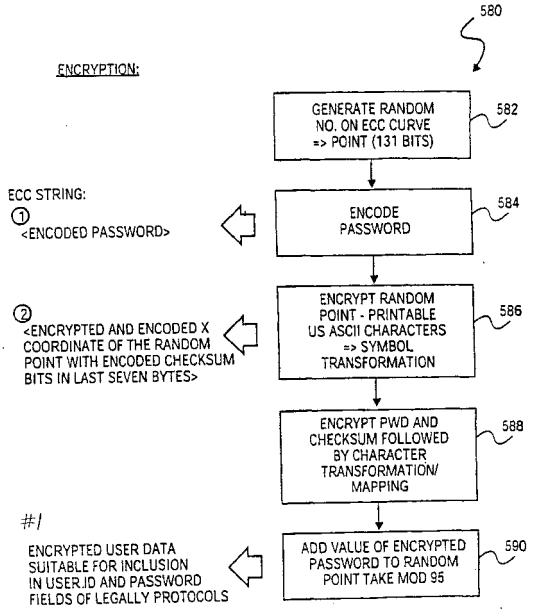
468 トランザクション・サーバ
552 暗号解読
554 受信したカウントを最後の
556 カウントと比較する
558 受信したアクセス・タイム
588 チェックサムを生成する
590 ダイヤラーから送信された
チェックサムと比較する
AUTHORIZATION REQUEST AND PLAIN TEXT PASSWORD: 認証要求および平文パスワード
574 トランザクション・サーバ・データベースを実行するために権利承認したカウント
CHECKSUM: チェックサム
COUNT DATA: カウント・データ
DIALER ID: ダイヤラーＩＤ
COUNT: カウント
MATCH: 一致する
REJECT: 拒絶する
454 ローム・サーバ

【図９】



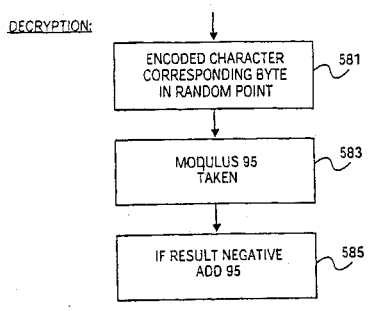
472　ローム・サーバ　　554　受信したカウントを最後の
552　ローム・サーバの暗号解読　　カウントと比較する
・ダイヤラー ＩＤ　　556　カウント・ジェネレータ
・最後のカウント　　→有効な要求
・最後のアクセス・タイム　　558　受信したカウントが最後の
・秘密鍵　　カウントと等しい→ウィン
ドウ期間中の最後のアクセ
588　チェックサムを生成する　　ス・タイムは？
590　ダイヤラーから送信された　　572　トランザクション・サーバ
チェックサムと比較する　　ＥＣＣ署名を計算および平文パスワード
AUTHORIZATION REQUEST AND PLAIN TEXT PASSWORD: 認証要求および平文パスワード　　576　トランザクション・サーバ・
データベースを更新するた
めに暗号解読したカウント
を追加する
574　認証応答
CHECKSUM: チェックサム
COUNT DATA: カウント・データ
DIALER ID: ダイヤラー ＩＤ
COUNT: カウント
RESULT: 指定する
MATCH: 一致するか

【図１０】



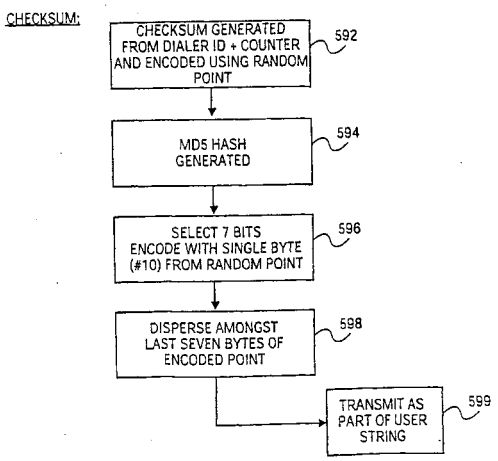
５８２　ＥＣＣ曲線上で乱数を生成する→点（１３１ビット）
５８４　パスワードを符号化する
５８６　ランダムな点を暗号化する－印刷可能なＵＳ　ＡＳＣＩＩ文字→記号変換
５８８　パスワードおよびチェックサムの暗号化，次いで文字変換／マッピング
５９０　ランダムな点に暗号化されたパスワードの値を加え、モデュラス９５を計算する
ENCRYPTION:　暗号化：　　ECC STRING:　ＥＣＣストリング：
①　＜符号化されたパスワード＞
②　＜最後の７バイトにおける符号化されたチェックサム・ビットを有する
ランダムな点の暗号化され符号化されたＸ座標＞
＃１　合法的プロトコルのユーザＩＤおよびパスワード・フィールドに
含めるのに適した暗号化されたユーザ・データ

【図１１】



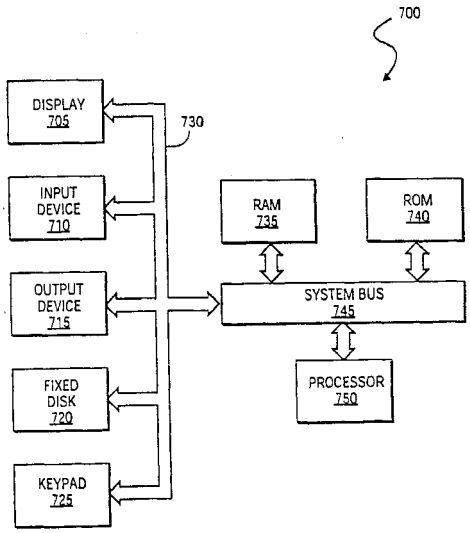
DECRYPTION:　暗号解読
５８１　ランダムな点のバイトに対応する符号化された文字
５８３　モデュラス９５が計算される
５８５　結果が負の場合は９５を加える

【図１２】



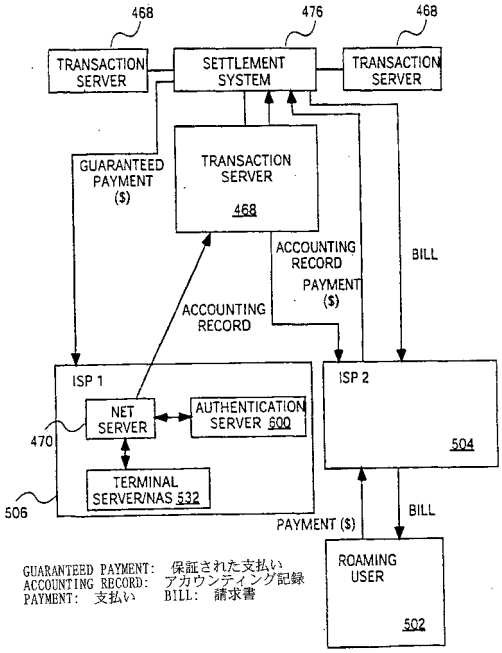
CHECKSUM:　チェックサム
５９２　チェックサムがダイヤラーＩＤとカウンターから生成され、
ランダムな点を使用して符号化される
５９４　ＭＤ５ハッシュが生成される
５９６　７ビットを選択し、ランダムな点の単一バイト（＃１０）で符号化する
５９８　符号化された点の最後の７バイトに分散する
５９９　ユーザ・ストリングの一部として送信する

【図１３】



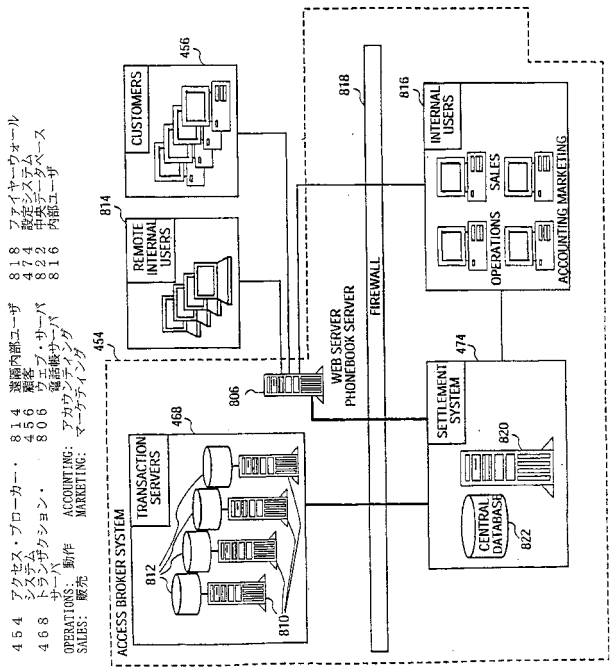
７０５　ディスプレイ
７１０　入力デバイス
７１５　出力デバイス
７２０　固定ディスク
７２５　キーパッド
７４５　システム・バス
７５０　プロセッサ

【図１４】



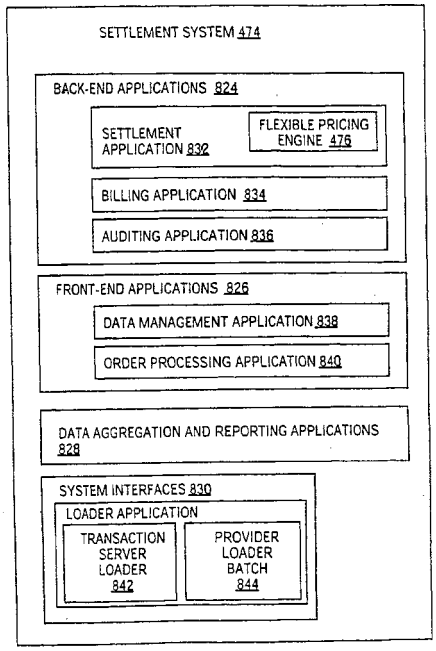
４６８　トランザクション・サーバ
４７６　設定サーバ
４７０　ネットサーバ
６００　認証サーバ
５３２　ターミナル・サーバ／ＮＡＳ
５０２　ローミング・ユーザ

【図１５】



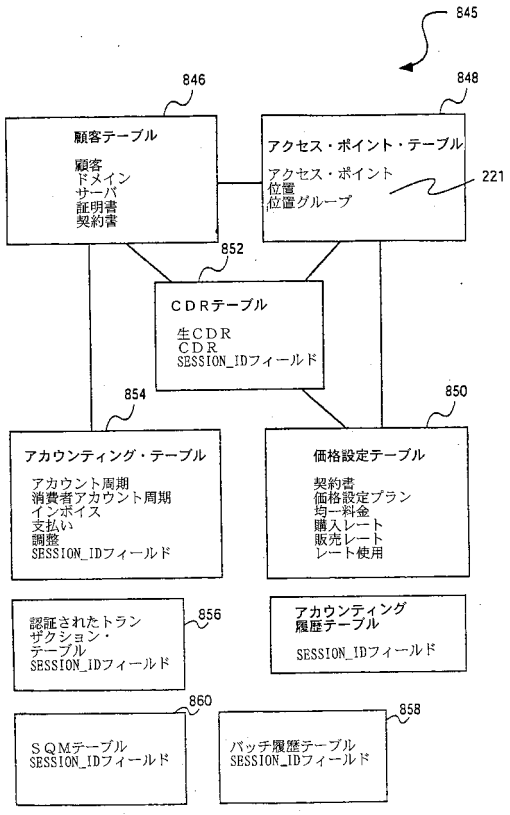
４５４　アクセス・ブローカー・システム・トランザクション・サーバ
４６８　サーバ
OPERATIONS:　動作
SALES:　販売
潤滑内部ユーザ　８１４
販売　４５６
ウェブ・サーバ　８０６
電話帳サーバ
ACCOUNTING:　アカウンティング
MARKETING:　マーケティング
８１８　ファイヤーウォール
４７４　設定システム
８２２　中央データベース
８１６　内部ユーザ

【図１６】

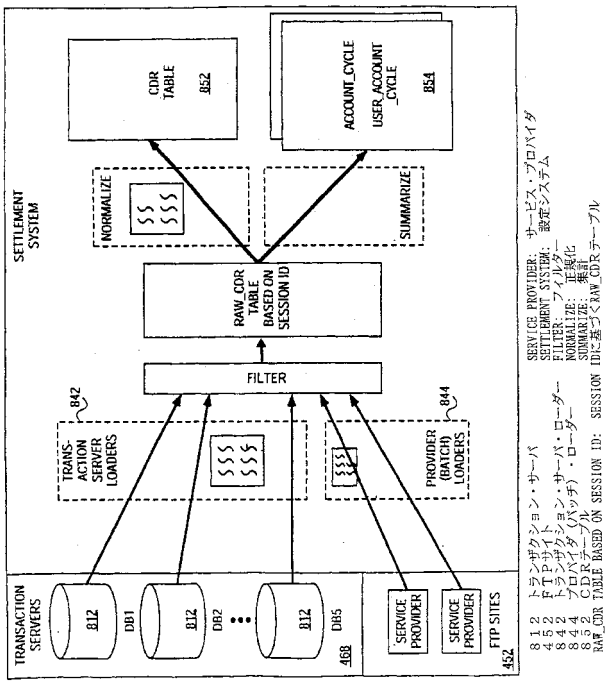


４７４　設定システム
８２４　バックエンド・アプリケーション
８３２　設定アプリケーション
４７６　柔軟な価格設定エンジン
８３４　請求書作成アプリケーション
８３８　監査アプリケーション
８２６　フロントエンド・アプリケーション
８３８　データ管理アプリケーション
８４０　注文処理アプリケーション
８２８　データ蓄積／報告アプリケーション
８３０　システム・インターフェース
８４２　トランザクション・サーバ・ローダー
８４４　プロバイダ・ローダー・バッチ
LOADER APPLICATION:　ローダー・アプリケーション

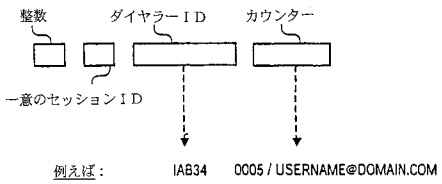
【図１７Ａ】



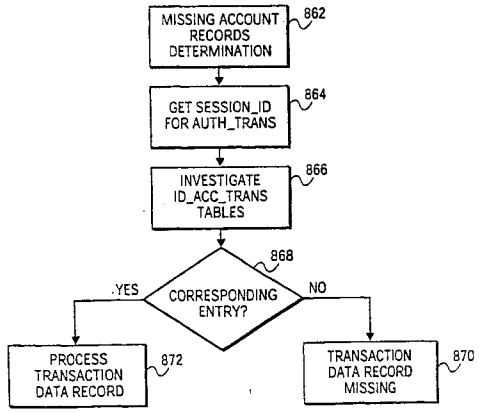
【図１７Ｂ】



【図１８】

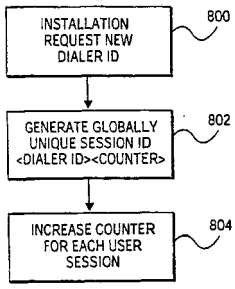


【図１９】



８６２　欠落したアカウント記録の決定
８６４　ＡＵＴＨ＿ＴＲＡＮＳのためにＳＥＳＳＩＯＮ＿ＩＤを取得する
８６６　ＩＤ＿ＡＣＣ＿ＴＲＡＮＳテーブルを調査する
８６８　対応する入力か？
８７２　トランザクション・データ記録を処理する
８７０　トランザクション・データ記録が欠落している

【図２０】



８００　新しいダイヤラーＩＤのインストール要求
８０２　大域的に一意のセッションＩＤ
　　<ＤＩＡＬＥＲ　ＩＤ><ＣＯＵＮＴＥＲ>
　　を生成する
８０４　ユーザ・セッションごとにカウンターを増やす

【国際公開パンフレット】

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
31 October 2002 (31.10.2002)

PCT

(10) International Publication Number
WO 02/086716 A1

(51) International Patent Classification[7]: G06F 01/24

(21) International Application Number: PCT/US02/12475

(22) International Filing Date: 18 April 2002 (18.04.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/284,914 18 April 2001 (18.04.2001) US
10/118,380 5 April 2002 (05.04.2002) US

(71) Applicant *(for all designated States except US)*: **IPASS, inc.** [US/US]; 3800 Bridge Parkway, Redwood City, CA 94065 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **EDGETT, Jeff, Steven** [US/US]; 151 South Bernado #24, Sunnyvale, CA 94086 (US). **SUNDER, Singam** [US/US]; 539 Isaac Court, San Jose, CA 95136 (US).

(74) Agents: **MALLIE, Michael, J.** et al.; Blakely, Sokoloff, Taylor & Zafman LLP, 7th floor, 12400 Wilshire Boulevard, Los Angeles, CA 90025 (US).

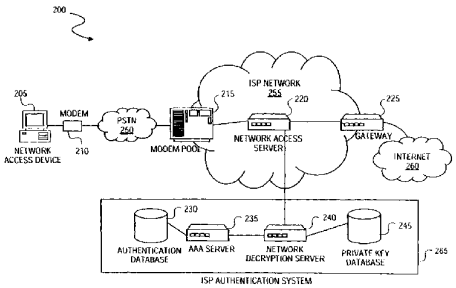
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

[Continued on next page]

(54) **Title:** METHOD AND SYSTEM FOR ASSOCIATING DATA RECORDS



(57) **Abstract:** A method (400) of, and system (200) for, associating a plurality of transaction data records generated in a service access system including a plurality of service providers (452) is provided. The transaction data records are generated in response to a user accessing the system during a single user session. The method includes generating a unique session identification (520) that is uniquely associated with the single user session and which is receivable by the service providers. The unique session identification (520) is included in the transaction data record. The plurality of transaction data records is received at a transaction processing facility from the service providers and processes using the unique session identification of each transaction data record. In certain embodiments, the unique session identification is provided in a user identification string of each transaction data record when the user session is authentication.

WO 02/086716 A1

WO 02/086716 A1

before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

METHOD AND SYSTEM FOR ASSOCIATING DATA RECORDS

FIELD OF THE INVENTION

[0001] The present invention relates generally to processing transaction records in a network access environment. More particularly, the present invention relates to a method of, and system for, associating a plurality of transaction data records generated in a service access system.

BACKGROUND

[0002] Due to the increasing globalization of economies, the need to provide communications between geographically dispersed persons and facilities has increased. For example, a particular business may have facilities located across multiple countries and continents. A further result of increased globalization has been an increase in business travel. The increasing dependence of corporations and persons on Internet-based communications has furthermore made it desirable that mobile workers (so-called "road warriors") be able to access Internet-based and wireless communications as they travel worldwide. Services that facilitate communications to such mobile persons are commonly referred to as "roaming services". Considering Internet-based communications as an example, in order to meet the needs of mobile customers, Internet Service Providers (ISPs) have begun to offer local-call access to the Internet from various locations world wide, such a service being termed a "roaming" Internet access solution. The requirement for a roaming solution arises primarily because ISPs tend to specialize by geographic area, causing gaps in service coverage. The expansion of network infrastructure, network management and continuous upgrades to meet required reliability and performance standards all place tremendous capital and time burdens on ISPs. For these reason, many ISPs only locate Points of Presence (POPs) in a limited geographic area.

[0003] For the reasons set out above, the ability for ISPs to offer Internet roaming solutions, especially to business customers, is becoming increasingly important as many businesses utilize Internet-based communications to replace

traditional remote access solutions for their telecommuters and mobile work forces.

[0004] In order to provide Internet roaming solutions, some ISPs have begun to share network infrastructure to gain additional geographic reach. This infrastructure sharing might take the form of an agreement to allow users of one ISP to gain Internet access through another ISP's network. It will be appreciated that, as the number of relationships between ISPs increase in an attempt to provide global coverage, managing and processing accounting information and sharing costs may become complex. For example, even with a relatively low number of ISPs, issues often arise relating to missing accounting records, inappropriate accounting records, duplicate accounting records, duplicate alias records, invalid session length records, overlapping accounting records, or the like.

[0005] Typically, users access the Internet through an Internet Service Provider (ISP). A network user attempting to gain access to the Internet or a corporate local area network (LAN) must generally enter a username and password for identification verification purposes. A problem with this process is that the password is generally not secure when transmitted to the ISP using many standard authentication protocols.

[0006] Figure 1 illustrates a diagram of a prior art ISP network configuration 100 in which network user credentials are authenticated using an insecure method. An ISP network 145 includes a network access server (NAS) 120 connected to a modem pool 115 and to the Internet 150 via a gateway 125. The ISP network 145 is also connected to an authentication server (AAA server) 135. The AAA server 135 may be local to the ISP network 145 or in a remote location a great distance from the ISP Network 145.

[0007] To establish an Internet connection, a network user typically executes a dial-up networking application on a network access device 105. The dial-up networking application prompts the network user to enter a network username and a network password and manipulates a modem 110 in order to initiate a modem session with the modem pool 115 over a public switched telephone network (PSTN) 140. After a modem session has been established, the dial-up

networking application begins communicating with the NAS 120 for purposes of establishing a data connection and authenticating the network user.

[0008] One of the more common data communication protocols used to establish connections between computers is the point-to-point protocol (PPP). One particularly well-known authentication protocol, which is commonly used in conjunction with PPP, is the Password Authentication Protocol (PAP). A dial-up networking application configured to use PAP repeatedly sends the username and password pair over the established data connection until an authorization acknowledgement signals is received or the connection is terminated. The dial-up networking application is configured to control the frequency and timing of the username and password transmission.

[0009] A problem with PAP is that the password is not encrypted before it is sent over the data connection, but instead, it is sent as clear text. This means that the password is susceptible to interception by a hacker. For example, a hacker with access to the data connection can use a network monitoring application to capture and display data packets that are sent across the data connection. Such network monitoring applications are common and are often referred to as packet sniffing or packet snooping applications due to their illicit use.

[00010] Referring again to Figure 1, once the username and password pair is received at the NAS 120, Remote Authentication Dial In User Service (RADIUS), another standard authentication protocol, is typically used to transmit the network username and password pair to an ISP authentication system 155. The RADIUS protocol provides for the symmetric encryption of the password before it is sent to the AAA server 135 in the ISP authentication system 155. The encryption method is considered symmetric because the NAS 120 and the AAA server 135 share a secret key used in the encryption algorithm. The NAS 120 uses the secret key to "lock", or encrypt, the password, while the AAA server 135 uses the secret key to "unlock", or decrypt, the password before checking the password against the password stored in an authentication database 130.

[00011] A problem with the RADIUS symmetric encryption method is that it is susceptible to a form of attack known as a "dictionary" attack. In a dictionary

attack, a hacker with knowledge of the encryption algorithm intercepts an encrypted password with a packet sniffing application. Then, the hacker repeatedly tries a series of keys until one is found that yields readable characters. To make matters worse, once the secret key is compromised, a hacker can readily decrypt any password intercepted between the NAS 120 and the AAA server 135.

[00012] In response to the weaknesses inherent in the PAP/RADIUS authentication method just described, the Challenge Handshake Authentication Protocol (CHAP) was developed. In a system implemented to use CHAP, the dial-up application in the network access device 105 negotiates with the NAS 120 to use CHAP as the authentication protocol, instead of PAP. Next, the NAS 120 generates a random number and sends it to the network access device 105. The dial-up networking application executing on the network access device 105 uses the random number to generate a non-reversible hash of the password, which is then sent to the NAS 120. The NAS 120 then uses the RADIUS protocol and sends the non-reversible hash and the random number used to generate the hash to the AAA server 135. The AAA server 135 retrieves the clear text password from the authentication database 130 and repeats the hash operation using the random number received from the NAS 120. Finally, the AAA server 135 compares its generated hash value with the hash value received from the NAS 120. If the hash values are the same, the authentication is considered successful and the AAA server 135 sends the appropriate acknowledgement signal to the network access device 105.

[00013] A problem with the CHAP/RADIUS method for user authentication is that all three systems, namely the network access device 105, the NAS 120 and the AAA server 135, must be configured to use CHAP in order to take advantage of the added security. If any of the three are not configured to use CHAP, the dial-up networking application on the network access device 105 uses the PPP protocol to negotiate with the NAS 120 to use PAP as the authentication protocol.

[00014] Another disadvantage of using the CHAP/RADIUS method is that in order for CHAP to be implemented properly, the AAA server 135 must have

access to clear text passwords. Many authentication systems do not store passwords in clear text form because of the added security risk that would result if the system were compromised and the passwords stolen.

[00015] More recently, authentication systems have deployed an authentication protocol referred to as Extensible Authentication Protocol (EAP). EAP works in much the same way as CHAP, except that the AAA server 135, not the NAS 120, generates the random number which the network access device 105 uses to hash the password. Consequently, EAP is subject to the same disadvantages of CHAP. Particularly, EAP is only effective if all systems in the authentication chain employ EAP.

[00016] With the advent of Broadband access, both wireless and wireline (ethernet) access providers employ web browser based authentication systems. The web browser uses Hyper Text Transport Protocol (HTTP) or Hyper Text Transport Protocol over Secure sockets layer (HTTPS) for transmitting the user credentials to the access point. A problem with HTTP is that the password is not encrypted before it is sent over the data connection, but instead, it is sent as clear text. This means that the password is susceptible to interception by a hacker. For example, a hacker with access to the data connection can use a network monitoring application to capture and display data packets that are sent across the data connection. Such network monitoring applications are common and are often referred to as packet sniffing or packet snooping applications due to their illicit use. A problem with HTTPS is that the access point needs to obtain the certificate from a well-known Certificate Authority (CA). This increases the cost of setting up the access point. The strength of the encryption used by HTTPS is regulated by government export restrictions. The web browsers include the weaker keys by default, and the users are expected to upgrade the encryption strength depending upon export restrictions. For the purposes of this specification, the term "connection application" should be construed as including, but not limited to, any device (both hardware and software) including functionality to authenticate data e.g., a peer-to-peer authentication arrangement, a dialer, a smart client, a browser, a supplicant, a

smart card, a token card, a PDA connection application, a wireless connection, an embedded authentication client, an Ethernet connection, or the like.

SUMMARY OF THE INVENTION

[00017] In accordance with the invention, there is provided a method of associating a plurality of transaction data records generated in a service access system including at least one service provider, the transaction data records being generated in response to a user accessing the system during a single user session and method including:

generating a unique session identification that is uniquely associated with the single user session and which is receivable by the at least one service provider, the unique session identification being included in the transaction data record;

receiving the plurality of transaction data records at a transaction processing facility from the at least one service provider; and

processing the transaction data records using the unique session identification of each transaction data record.

[00018] Still further in accordance with the invention, there is provided a system for processing transaction data records generated in a service access system including at least one service provider, the transaction data records being generated in response to a user accessing the system during a single user session and system including:

a session identification generator to generate a unique session identification that is uniquely associated with the single user session and which is receivable by the at least one service provider, the unique session identification being included in the transaction data record; and

a transaction processing facility to process the plurality of transaction data records received from the at least one service provider using the unique session identification of each record.

[00019] In accordance with a further aspect of the invention, there is provided a method of processing a plurality of transaction data records generated in a service access system including at least one service provider, method including:

receiving the transaction data records from the at least one service provider, each transaction data record being generated in response to a user accessing the system during a single user session; and

identifying transaction data records associated with the single user session based on a session identification included in each transaction data record, each session identification uniquely identifying a single user session.

[00020] In accordance with a yet further aspect of the invention, there is provided a transaction processing facility for processing a plurality of transaction data records generated in a service access system including at least one service provider, transaction processing facility arranged to:

receive the transaction data records from the at least one service provider, each transaction data record being generated in response to a user accessing the system during a single user session; and

identify transaction data records associated with the single user session based on a session identification included in each transaction data record, each session identification uniquely identifying a single user session.

[00021] In accordance with a yet further aspect of the invention, there is provided a method of connecting a user to an access service provider, the method including creating a unique session identification associated with a single user session during which the user accesses the service provider, the unique session identification being provided in a user identification string of each transaction data record when the user session is authorized.

[00022] The invention extends to a machine-readable medium embodying a sequence of instructions for carrying any one of the methods herein described.

[00023] Still further in accordance with the invention, there is provided a transaction processing facility for processing a plurality of transaction data records generated in a service access system including at least one service provider, transaction processing facility including:

receiver means for receiving the transaction data records from the at least one service provider, each transaction data record being generated in response to a user accessing the system during a single user session; and

processor means to identify transaction data records associated with the single user session based on a session identification included in each transaction data record, each session identification uniquely identifying a single user session.

[00024] Other features and advantages of the present invention will be apparent from the drawings and detailed description that follow.

BRIEF DESCRIPTION OF THE DRAWINGS

[00025] The present invention is illustrated by way of example, and not intended to be limited by the figures of the accompanying drawings, in which like references indicate the same or similar elements and in which:

[00026] Figure 1 is a diagram of a prior art ISP network configuration in which network user credentials are authenticated using an insecure method;

[00027] Figure 2 is a diagram of a network configuration including an ISP network, a network access device, and a network decryption server, consistent with an embodiment of the invention;

[00028] Figure 3 is a diagram of a network configuration including a remote ISP network, a network access device and a network decryption server, consistent with an embodiment of the invention;

[00029] Figure 4 is an exemplary flow diagram of the operations for a method to securely authenticate network user credentials;

[00030] Figure 5 is a block diagram of a multi-party service access environment, in accordance with an exemplary embodiment of the invention, which includes a number of service providers, an access broker system and multiple customers;

[00031] Figure 6 is a schematic diagram illustrating operation of an access broker system, in accordance with an exemplary embodiment of the invention, to provide roaming Internet access;

[00032] Figure 7 is a schematic functional block diagram of a connect dialer in accordance with a exemplary embodiment of the invention;

[00033] Figure 8 is a schematic functional block diagram of a transaction server, in accordance with an embodiment of the invention, which includes decryption functionality;

[00034] Figure 9 is a schematic functional block diagram of customer or roam server, in accordance with another embodiment of the invention, which includes decryption functionality;

[00035] Figure 10 is a schematic flow diagram of an exemplary encryption method performed by the connect dialer;

[00036] Figure 11 is a schematic flow diagram of an exemplary decryption method performed by the transaction server or customer server;

[00037] Figure 12 is a schematic flow diagram of an exemplary encryption method of checksum data;

[00038] Figure 13 is a schematic diagram of a computer system, which may be configured as a network access device or a network decryption server.

[00039] Figure 14 is a schematic block diagram illustrating operation of an access broker system to provide roaming Internet access, in accordance with one embodiment of the invention;

[00040] Figure 15 is a schematic block diagram of exemplary physical architecture of the access broker system of Figure 14;

[00041] Figure 16 is a schematic block diagram of an exemplary settlement system;

[00042] Figure 17A shows an exemplary data model used in the access broker system;

[00043] Figure 17B is a schematic diagram illustrating processing, in accordance with the invention, using a unique session identification also in accordance with the invention;

[00044] Figure 18 shows an exemplary unique session identification in accordance with one embodiment of the invention;

[00045] Figure 19 shows a schematic flow chart of methodology to identify missing transaction data records using a unique session identification; and

[00046] Figure 20 shows a schematic flow chart of unique session identification methodology at a connection application also in accordance with an embodiment of the invention.

DETAILED DESCRIPTION

[00047] A method and system for securely authenticating network user credentials or user data are described. A network access device encrypts a network user credential, such as a password, input by a network user. The network access device encrypts the network user credential with a public key, which is part of a public/private key pair, generated with a strong encryption algorithm. The network access device transmits the encrypted network password to a network decryption server. The network decryption server decrypts the network user credential using the private key of the public/private key pair. The network decryption server transmits the decrypted password to an authentication (AAA) server for verification. If the password is positively verified at the AAA server, the AAA server sends an appropriate acknowledgment signal to the network access device indicating that the password has been properly verified or authenticated. Based on the acknowledgement signal, the network access device gains access to the Internet or some other resource.

[00048] By encrypting the network password at the network access device with an asymmetric public key based on a strong encryption algorithm, the password can be securely transmitted from the network access device to a network decryption server. If the encrypted password is captured by a sniffing or snooping application at some point between the network access device and the network decryption server, the encrypted password can only be decrypted with knowledge of the correct private key and the encryption algorithm. Preferably, decryption of the user credentials takes place as close as possible to the source which the user wishes to access.

[00049] The embodiment of the invention depicted in the drawings is independent of the underlying authentication protocols and therefore can be implemented to work with a variety of new and existing authentication protocols. Moreover, this embodiment of the invention provides for secure authentication while resolving the need to fully standardize the capability of the authentication chain. For example, by passing encrypted data through standard PPP/RADIUS information fields, the invention provides a secure

authentication method without the hassle and expense of implementing and configuring network equipment to work with more complex authentication protocols, such as CHAP and EAP. It is, however, to be appreciated that the invention may be used with CHAP, EAP and other protocols and is not limited to application in a PAP/RADIUS environment.

[00050] Figure 2 is a diagram of a network configuration 200 including an ISP network 255, a network access device 205 and a network decryption server 240, consistent with one embodiment of the invention. The ISP network 255 includes a NAS 220, a modem pool 215 and a gateway 225. The ISP network 255 is connected to the Internet via the gateway 225 and connected to an ISP authentication system 265 via a connection between the NAS 220 and a network decryption server 240. In one embodiment, the ISP network 255 and the ISP authentication system 265 are physically located within the same facility. However, in an alternative embodiment, the ISP authentication system 265 is located in one facility and connected via a wide area network (WAN) to one or more ISP networks, such as the ISP network 255. This type of configuration allows for the individual ISP networks to be strategically located in different geographical areas thereby allowing customers to access the network via a local telephone call, while centralizing the authentication system for added security.

[00051] In one embodiment of the invention, to access the Internet 260, a network user executes a dial-up connection application on the network access device 205. In alternative embodiments, other types of network connection applications may be utilized to access the Internet. The dial-up connection application prompts the network user to input a network username and a network password and manipulate a modem 210, causing it to establish an audio communication session with the modem pool 215. Although the modem 210 is shown in Figure 2 as an external device, in alternative embodiments of the invention, the modem 210 may be an internal device integrated with the network access device 205. Once an audio communication session has been established, the NAS 220 begins communicating with the network access device 205 for the purpose of authenticating the network user.

[00052] Before the network access device 205 sends the network credentials entered by the network user, the network password is encrypted. The password is encrypted using the public key of a public/private key pair. This encryption technique is well known in the art and is generally referred to as asymmetric public key cryptography. In asymmetric public key cryptography, a person makes one key publicly available and holds a second, private key. A message is "locked", or encrypted, with the public key, sent, and then "unlocked", or decrypted, with the private key.

[00053] In the embodiment of the invention depicted in the drawings, a strong encryption algorithm is used to generate the public/private key pair. The public key and private key have a mathematical relationship based on an elliptic curve. This encryption technique is well known in the art and is generally referred to as elliptic curve cryptography or ECC. Public key encryption algorithms rely on a one-way mathematical problem, which makes it easy to generate a public key from a private key but difficult to deduce the private key, given the public key. Elliptic curve systems use an algebraic formula to determine the relationship between public and private keys within the universe created by an elliptic curve. Elliptic curve cryptography is advantageous because the key sizes are small relative to other strong encryption techniques. This allows a password to be encrypted with strong encryption and yet, an encrypted password still fits in the password data field defined by the popular authentication protocols, such as PAP, CHAP, EAP, and RADIUS.

[00054] Referring again to Figure 2, the public key is known to the network access device 205, while the private key is stored in a private key database 245. The network access device 205 encrypts the password using the public key before sending the network username and the encrypted network password to the NAS 220. The NAS 220 forwards the network username and the encrypted network password to the network decryption server 240. The network decryption server 240 uses the network username as an index into the private key database 245 and retrieves the private key associated with the network username. Then, the network decryption server 240 uses the private key to

decrypt the encrypted network password and to generate the original clear text password as input by the network user.

[00055] Finally, the network decryption server 240 forwards the network username and the clear text network password to the AAA server 235 for verification. The AAA server 235 uses the network username as an index into the authentication database 230 to retrieve the official password that is associated with the network username. If the official password matches the password input by the network user and sent by the network access device 205, the AAA server 235 sends an appropriate acknowledgment signal to the NAS 220, and the NAS 220 forwards the signal to the network access device 205, acknowledging the successful verification and granting access to the Internet or some other resource.

[00056] One embodiment of the invention is independent of the authentication protocols used to send the credentials from the network access device 205 to the NAS 220 and ultimately to the AAA server 235. For example, the invention can be implemented to work with popular authentication protocols such as PAP, CHAP, EAP and RADIUS, among others.

[00057] For one embodiment of the invention, the NAS 220 is configured to use PAP and RADIUS for authenticating network user credentials. When configured for PAP/RADIUS, the NAS 220 negotiates the use of PAP with the network access device 205 when the communication session between the NAS 220 and the network access device 205 is initiated. The NAS 220 is configured as a RADIUS client of the AAA server 235, which is a RADIUS server. The network decryption server 240 is also configured as a RADIUS server, but acts as a RADIUS proxy client to the AAA server 235. In this configuration, the network access device 205 encrypts the password, as entered by the network user. Then, the network access device 205 creates a PAP packet and places the network username and encrypted network password into the proper fields within the packet. Next, the network access device 205 sends the PAP packet to the NAS 220. The NAS 220 forwards the data to the network decryption server 240 using a RADIUS packet. The network decryption server 240 decrypts the

password and uses RADIUS to forward the clear text password to the AAA server 235 for verification.

[00058] In an alternative embodiment, the NAS 220 is configured to use CHAP and RADIUS to authenticate network user credentials. In a network configured to use CHAP/RADIUS, the NAS 220 negotiates with the network access device 205 to use CHAP as the authentication protocol, instead of PAP. Next, the NAS 220 generates a random number and sends it to the network access device 205. The dial-up connection application executing on the network access device 205 uses the random number to generate a non-reversible hash of the password using a pre-determined encryption algorithm. Rather than encrypt the actual password, the network access device 205 encrypts the non-reversible hash of the network password in accordance with the exemplary embodiment of the invention as described above. The network access device 205 creates a CHAP packet and sends the network username and the encrypted non-reversible hash to the NAS 220.

[00059] The NAS 220 sends the data, including the network username, the encrypted non-reversible hash, and the original random number used to generate the non-reversible hash, to the network decryption server 240 using the RADIUS protocol. The network decryption server 240 decrypts the non-reversible hash and replaces the non-reversible hash in the RADIUS packet, which is forwarded to the AAA server 235.

[00060] The AAA server 235 receives the packet and retrieves the password associated with the network username from the authentication database 230. The AAA server 235 uses the random number originally generated at the NAS 220 to perform a hash operation on the original password retrieved from the authentication database 230. Next, the AAA server 235 compares the hash it generated to the hash it received from the network access device 205. If the two hashes match, the verification is successful and the AAA server 235 sends an appropriate acknowledgment signal to the network access device 205 granting access to the Internet 260 or some other resource.

[00061] In another embodiment of the invention, the NAS 220 is configured to use EAP and RADIUS. EAP works in much the same way as CHAP, except the

random number sent to the network access device 205 is generated by the AAA server 235 instead of the NAS 220. Because the invention works with any authentication protocol, the invention can easily be implemented to work with a variety of network configurations and provides a very strong, minimal level of security using LEGACY systems.

[00062] Figure 3 is a diagram of a network configuration 300 including a remote ISP network 365, a network access device 305 and a network decryption server 350, consistent with one embodiment of the invention. The remote ISP network 365 includes a NAS 320, a modem pool 315 and a gateway 325. The remote ISP network 365 is connected to the Internet 370 via the gateway 325 and connected to a remote ISP authentication system 375 via a connection between the NAS 320 and the AAA server 335. The remote ISP authentication system 375 is connected to a local ISP authentication system 380 via a WAN connection between the AAA server 335 and the network decryption server 350.

[00063] The configuration 300 allows a network user via the network access device 305 to accesses the Internet 370 through the remote ISP network 365. A local ISP, which operates and maintains the local ISP authentication system 380, makes arrangements with a remote ISP, such that network users of the local ISP are allowed access to the Internet via the remote ISP network 365, which is maintained and operated by the remote ISP. This type of business arrangement might exist where the remote ISP is located in a distant geographical area or different country from the local ISP. The embodiment of the invention depicted in Figure 3 is particularly advantageous in this type of configuration because of the inability of the local ISP operators to control who has access to the equipment that comprises the remote ISP network 365 and the remote ISP authentication system 375. Further, the remote ISP network 365 only has access to an encrypted version of the password thereby enhancing security.

[00064] The embodiment of the invention illustrated in Figure 3 works in much the same way as discussed above in relation to Figure 2, except that the encrypted password passes through the remote ISP network 365 and the remote ISP authentication system 375. Referring to Figure 3, to access the Internet 370, a network user executes a dial-up connection application on the network access

device 305. The dial-up connection application prompts the network user to input a network username and network password and manipulates the modem 310, causing it to establish an audio communication session with the modem pool 315. Once an audio communication session has been established, the NAS 320 begins communicating with network access device 305 for the purpose of authenticating the network user.

[00065] Before transmitting the network password to the NAS 320, the network access device 305 encrypts the network password with a public key as discussed above. The network access device 305 then creates a data packet destined for the local ISP authentication system 380 and forwards the packet to the NAS 320 of the remote ISP 365. The NAS 320 receives the data packet containing the encrypted password and forwards it to the remote ISP authentication system 375 and the AAA server 335 in particular. The AAA server 335 examines the data packet, discovers it is destined for the local ISP authentication system 380, and forwards the data packet to the network decryption server 350.

[00066] The network decryption server 350 receives the data packet and retrieves the private key associated with the network username from a private key database 355. Then, the network decryption server 350 decrypts the encrypted password and forwards the data packet with the clear text password to the AAA server 345 for verification. The AAA server 345 uses the network username as an index into the authentication database 340 to retrieve the clear text password associated with the username from the authentication database 340. If the retrieved password matches the password received from the network access device 305, then the AAA server 345 sends an appropriate acknowledgment signal to the AAA server 335 of the remote ISP 365. The AAA server 335 forwards the signal to the NAS 320. The NAS 320 forwards the signal to the network access device 305 acknowledging the successful verification and granting access to the Internet or some other resource. Thus, decryption takes place in proximity to a local ISP associated with the user and any one or more intermediary ISPs only have access to encrypted authentication data.

[00067] Figure 4 illustrates a flow diagram of the operations 400 for a method to securely authenticate network user credentials, consistent with one embodiment of the invention. The method begins at operation 405. At operation 405, a network access device uses a public key, which is part of a public/private key pair, to encrypt a network credential such as a password. The public/private key pair has been previously generated based on a strong encryption algorithm, such as elliptic curve cryptography.

[00068] At operation 410, the network access device transmits the encrypted network credential to a network decryption server. The encrypted password may be forwarded through several network nodes, including network access servers and AAA servers before it ultimately reaches the network decryption server.

[00069] At operation 415, the network decryption server decrypts the encrypted network credential using the private key of the public/private key pair referred to above. The network decryption server may retrieve the private key from a private key database, using the username as an index into the private key database.

[00070] Finally, at operation 420, the network decryption server transmits the decrypted network credential to an AAA server for verification. The decrypted network credential may be forwarded over several network nodes, such as network access servers or other AAA servers before ultimately reaching the AAA server for verification.

[00071] A typical application of the invention is in a multi-party service access environment and its application therein is described below. Such applications typically include roaming users, multiple service providers and multiple customers. Further, such applications typically use PAP, CHAP, EAP, RADIUS or the like protocols which communicate user credentials in an insecure fashion. However, the embodiment described below allows secure authentication in LEGACY systems.

Terminology

[00072] For the purposes of the present specification, the term "service access transaction" includes any transaction between a service customer and a service

provider for a user session. An example of such a service may be access to any communications network via any medium or protocol. For example, the communications networks may comprise packet-switched networks, circuit-switched networks, cable networks, satellite networks, terrestrial networks, wired networks, or wireless networks. The term "service access transaction", however, is not limited to a network access transaction, and encompasses a transaction pertaining to access to any one of a number of other services such as content, commerce and communications services.

[00073] For the purposes of the present specification, the term "customer" includes any entity involved in the purchase and/or consumption of service access, regardless of whether the service access is performed by the customer or not. For example, a "customer" may be an end-user consumer that actually utilizes the service access, or a corporate entity to which such an end-user belongs, an Internet service provider, an Internet carrier, a reseller, or a channel.

Multi-party services access environment

[00074] This embodiment of the present invention discloses a multi-party access broker and settlement system for service access (e.g., Internet access, content access, commerce access, or communications access) services that enable a service provider (e.g., an ISP, a wireless service provider, a VPN service provider, a content distribution service provider, an e-commerce service provider or an application service provider) to offer relatively secure service access in a multi-party access environment using standard communication protocols (e.g., PPP, HTTP) and standard authentication protocols (e.g., RADIUS, PAP, EAP or the like). Such protocols typically define a user field of a maximum length and the exemplary embodiment of the invention describes, inter alia, a method and system to provide secure authentication within a field with the abovementioned maximum length. Accordingly, the invention may be applied to LEGACY systems.

Overview

[00075] Figure 5 is a block diagram of an exemplary multi-party service access environment 450, in the exemplary form of a network access environment, which includes a number of service providers 452, an access

broker system 454, and multiple customers (or consumers) 456. At a high level, the service providers 452 have service (e.g., access, content, e-commerce services etc.) capacity that is sold, via the access broker system 454, to the multiple customers 456. Accordingly, the access broker system 454 may be regarded as purchasing service capacity (e.g., service access), which is then resold to the customers 456. While the service to which access is provided below is network access, it will be appreciated that access is described below as an exemplary service and, for the purposes of this specification should be taken to include any form of access as described above. In the exemplary embodiment, the service providers 452 may include any communication network service providers, such as ISPs 458 (e.g., UUNet Technologies, Genuity, CompuServe Network Services, EQUANT, Hong Kong Telecom, etc.), wireless access providers 460 (e.g., Verizon, Sprint, Pacific Bell), content distribution providers 462 and e-commerce providers 464. The service providers 452 may, however, include any number or type of service providers providing any number of services (e.g., access, content, communications or e-commerce services, to name but a few).

[00076] The exemplary access broker system 454 includes a number of components. A connection application is a client application typically in the form of a dial-up application or connect dialer 466, installed on a service or network access device (e.g., a computer system such as the access devices 205, 305 in Figures 2 and 3) of a customer 456 that facilitates convenient access to a communications network. In one embodiment, the connect dialer 466 may provide a simple point-and-click interface for dialing into a worldwide connection network of the access broker system 454. To this end, the connect dialer 466 may store multiple phone numbers for multiple ISPs worldwide with potentially different setup and dial-up scripting information. As described above broadly with respect Figures 1 to 4, the connect dialer 466 encrypts user data and counter data in such a fashion so that it may be included in the user string permitted or allowed by known protocols such as Point-to-Point protocol (PPP), Password Authentication Protocol (PAP), Challenge-Handshake Authentication Protocol (CHAP), Remote Authentication Dial In User Service

(RADIUS) protocol, Terminal Access Controller Access Control System (TACACS) protocol, Lightweight Directory Access Protocol (LDAP), NT Domain authentication protocol, Unix password authentication protocol, HyperText Transfer Protocol (HTTP), HyperText Transfer Protocol over Secure sockets layer (HTTPS), Extended Authentication Protocol (EAP), Transport Layer Security (TLS) protocol, Token Ring protocol and/or Secure Remote Password protocol (SRP).

[00077] The environment 450 also includes a plurality of transaction servers 468 that provide trusted third-party functionality of routing and logging user identification information, authorization responses and usage, and accounting information. The transaction servers 468 include decryption functionality and may thus define decryption servers.

[00078] Whereas the connect dialer 466 is installed on a client or user network access device 205, 305, the netservers 470 are installed at a "remote" ISP allowing its POPs to be utilized by roaming users, and roam servers 472 reside at a "home" ISP to allow a roam user access an associated home network. It should be noted that the transaction servers 468 operate to route messages between the network and roam servers 470 and 472.

[00079] A settlement system 474, including a flexible pricing engine 476, performs financial settlement of service access transactions between the service providers 452 and the customers 456. The access broker system 454 is also includes a Service Quality Monitor 478 (SQM) that facilitates the collection and analysis of quality of service (QoS) information for services provided to customers 456 and a phonebook management system 480 that facilitates management of multiple connect dialers 466 used by customers 456. The transaction servers 468 are accessed by the settlement system 474 to load transaction data. The various components in the environment 450 may include aspects of known functionality and, dependent upon the specific embodiment of the invention, certain components may be omitted.

The Customers

[00080] The customers 456, in the embodiment depicted in the drawings, are arranged in a multi-tier customer structure, whereby the access broker system

454 may interact with customers 456 that operate according to a variety of business plans and needs. At one end of the spectrum, the customer 456 may comprise an individual end-user that subscribes to a roaming system facilitated via the access broker system 454. Alternatively, the customer 456 may be in the form of a corporate customer 482 (e.g., a corporation or business) that purchases roaming Internet access for employees of the corporation.

[00081] Each customer 456 may also comprise an ISP customer 484 that purchases roaming Internet access for resale to its customers (e.g., end-users 486 and corporate customers 482). Each customer 456 may also operate as a solution partner or reseller 488 that markets and resells roaming Internet access brokered by the access broker system 454 to end-users 486, corporate customers 482 and/or ISP customers 484.

[00082] The customers 456 may also include parties regarded as Internet Carriers 490 (e.g., IXCs, RBOs, CLECs, ILECs and ISPs). It will thus be appreciated that in the multi-party access environment 450 a number of different service providers may participate in providing access to a roaming user and, accordingly, customer security issues and, in particular, secure authentication of users, are of importance. Also, as the number of participants increases, accounting issues tend to become more complex.

Roaming Service Access

[00083] Referring in particular to Figure 6, reference numeral 500 generally indicates an example of how the access broker system 454 may provide roaming Internet access in a relatively secure manner in accordance with an embodiment of the invention. When a roaming user 502, shown to be a subscriber to a "home" ISP 504, connects to a remote ISP 506 that provides a local POP 508 within a specific geographic area 510, the roaming user 502 inputs the same user name 512 and password 514 (i.e., authentication data or user credentials) used when connecting via a POP 509 of the "home" ISP 504. However, standard or LEGACY multi-party access environments typically use PAP for dialup authentication and HTTP POST based authentication for wired and wireless broadband authentication. This results in the passwords being transported via insecure media and their confidentiality may be compromised

and subsequently used to fraudulently access both networks of the access broker system 454 and the customers 456. In order to alleviate this problem, in accordance with one embodiment of the invention, user data is encrypted by the connect dialer 466 prior to communicating it to the POP 508, as described above with reference to Figures 1 to 4, and in the context of a multi-party environment with reference to Figures 5 to 13.

[00084] In the embodiment depicted in the drawings, the customers 456 use a web form for requesting the connect dialer 466. This web form may include fields that can be used for specifying the required customizations. For example, the following fields are included in the web form for Secured Password Authentication in Plain-text (hereinafter referred to as "Secure PAP") -

Enable Secure PAP encryption: (Y/N)

Public Key: ****

Key Id: (0-9)

[00085] When a customer 456 wants to enable Secure PAP for their roaming users 502 (see Figure 6), they use an ECC utility that is included in their associated roam server 472. The roam server 472 runs an application supplied by the access broker system 454 to the customers 456 and generates a public/private key pair. The private key is typically added to an esp_key_pair.txt file. The public key is typically sent to the dialer support team of the access broker system 454 using an appropriate form. The dialer support team uses a dialer customization tool (DCT) to build the connect dialer 466 in accordance with one embodiment of the invention. The DCT tool includes a web page for specifying the encryption/decryption algorithm to be used and the ECC public/private keys.

[00086] The connect dialer 466 maintains a dialer id and counter (see block 520 in Figure 7) for generating a unique session id (see block 522) that uniquely identifies a user access session. The connect dialer 466 may, for example, obtain the dialer id from a web server of the access broker system 454 and, in use, the connect dialer 466 increments the counter for each dial attempt so that each user access session is uniquely identified. The dialer id and a value of the counter are used to create the unique session id prefix. In order to ensure the integrity

of the dialer id and value or count of the counter (which are transmitted in the clear), these fields are used to generate a checksum character. The algorithm used for generating this checksum character is described in more detail below with reference to Figure 12. An exemplary embodiment of the unique session id is described in more detail later in this document.

[00087] The netserver 470 maintains a cache of authenticated user ids and passwords for a limited period so that subsequent authentications can be performed from the cache (see block 538). Since the secure PAP changes the user id and password for each authentication, it breaks any caching feature at the netserver 470. Thus, in certain embodiments, in order to maintain compatibility with the standardized netserver cache, the dialer 466 may store a random point locally and reuse this for limited period of time (see block 540). After the aforementioned processing, the netserver 470 communicates the authentication data to the transaction server 468.

[00088] Referring in particular to Figure 8, reference numeral 550 generally indicates exemplary functionality carried out by the transaction server 468. The transaction server 468 maintains the dialer id, the last used value of the counter and the last access time in a table (see block 552). This table is used for protecting the network against replay attacks. This table is typically replicated across all transaction servers 468.

[00089] Upon receipt of the user credentials or authentication data from the netserver 470, in one embodiment of the invention, the transaction server 468 decrypts the password, and compares the received value of the counter with the value in stored in its database (see block 554). If the count sent by the dialer 466 is greater than the last count value stored in the database, then it is considered a genuine request (see block 556). If the count sent by the dialer 466 is equal to the last count value stored in the database, and the delta or time difference between the current system time and the time of last access stored in the database is less than a time window allowed, then again the request is considered genuine (see block 558). The transaction server 468 rejects the authorization request as a possible replay attack if the count sent by the dialer 466 fails these two conditions (see block 560). The transaction server 468 sends

the authentication request along with the plain text password to the roam server 472 of Figure 9.

[00090] In the embodiment depicted in Figure 8 the transaction server 468 maintains a record of the customer's private key and, accordingly, decryption of the authentication data takes place at the transaction server 468, which may thus define a decryption server. However, certain customers may wish to not provide their private key to any intermediaries such as the transaction servers 468. In these circumstances, the customer's private key is not provided to the transaction servers 468 but rather to the customer's roam server 472 that is typically at an in-house location. Accordingly, in addition or instead, decryption of the authentication data may thus take place in a similar fashion to that described above at the customer's roam server 472. An embodiment of a roam server 472 that includes encryption functionality is shown in Figure 9.

[00091] Referring in particular to Figure 9, reference numeral 570 generally indicates exemplary functionality carried out by the roam server 472. As the functionality substantially resembles the functionality 550 of Figure 8, like reference numerals have been used to indicate the same or similar features. When the transaction server 468 does not have access to the particular customer's private key, the transaction server 468 adds the necessary ECC attributes to the authentication request packet and sends it to the roam server 472 (see block 572). The roam server 472 decrypts the password and the checksum character using the ECC information and the private key stored locally (see block 552). The roam server 472 then performs the same functionality tests described above to determine if the count is valid (see blocks 554 – 560). The roam server 472 adds the decrypted count to the authentication reply packet (see block 574) so that the transaction server 468 can update its database with the latest value of the count (see block 576). Exemplary tables for implementing counter functionality are set out below.

[00092] A table dialer_counter_ts is typically used for replication. This table is created at each Transaction Server 468.

TABLE: DIALER_COUNTER_TS

FIELD NAME	DESCRIPTION
DIALER_COUNTER_TS_ID	A NUMERIC ID. REQUIRED FOR ORACLE SNAPSHOTS.
SERVER_ID	THE TRANSACTION SERVER ID. VARCHAR2(20).
DIALER_ID	THE DIALER ID IS OBTAINED FROM THE DIALER_ID SERVLET AT A WEB SERVER OF THE SYSTEM 454. VARCHAR2(10)
COUNTER	LAST USED VALUE OF THE COUNTER. VARCHAR2(5)
ACCESS_TIME	LAST ACCESS TIME

[00093] The last used value is typically stored in a database instance e.g., on "SESSION" machine. The SESSION machine is typically used to pull the entries from dialer_counter_ts tables in the transaction servers 468 and aggregate them into a single table. The SESSION machine also creates a unique snapshot corresponding to every dialer_counter_ts table in the transaction servers 468. These snapshots are typically named as dialer_counter_ts_<ServerId>, where ServerId is the id of the particular transaction server 468. The exemplary database instance SESSION is created with two identical machines on either coast to enhance fault tolerance.

TABLE: DIALER_COUNTER	
FIELD NAME	DESCRIPTION
DIALER_ID	THE DIALER ID IS OBTAINED FROM THE DIALER_ID SERVLET AT A SYSTEM WEB SERVER OF THE SYSTEM 454 AND IS USED FOR UNIQUELY IDENTIFYING THIS RECORD. VARCHAR2(10)
COUNTER	LAST USED VALUE OF THE COUNTER. VARCHAR2(5)

ACCESS_TIME	LAST ACCESS TIME

[00094] Each transaction server 468 typically replicates the dialer_counter table using Oracle snapshots. When a standard system is upgraded to accommodate the present embodiment of the invention, the following exemplary modifications are typically made.

TABLE: SECURE_PAP	
FIELD NAME	DESCRIPTION
SPAP_ID	GENERATED ID THAT UNIQUELY IDENTIFIED THIS RECORD.
CUSTOMER_ID	CUSTOMER ID.
PUBLIC_KEY	PUBLIC KEY.
PRIVATE_KEY	PRIVATE KEY VALUE.
KEY_VERSION	KEY VERSION NUMBER.
ALGORITHM	ALGORITHM. FOR EXAMPLE, E AND A.
EXPIRATION_DATE	TIME/DATE WHEN THIS RECORD WILL EXPIRE. IF NULL, THIS RECORD WILL NEVER EXPIRE.
DESCRIPTION	DESCRIPTION ENTERED FROM DCT.
CREATION_DATE	TIME/DATE WHEN RECORD WAS CREATED.
MODIFY_BY	USER WHO MODIFIED RECORD.
MODIFY_TIME	TIME WHEN RECORD WAS MODIFIED.

TABLE: CUSTOMER	
FIELD NAME	DESCRIPTION
ENCRYPT_FLAG	0 = ENCRYPTION IS OPTIONAL, 1 = ENCRYPTION IS REQUIRED FOR THIS CUSTOMER

TABLE: DIALER_PROFILE	
FIELD NAME	DESCRIPTION
ENCRYPT_FLAG	0 = ENCRYPTION OFF, 1 = ENCRYPTION ON
SPAP_ID	REFERENCES SECURE_PAP TABLE

Encryption/Decryption functionality

[00095] In the embodiment described above with reference to Figures 7 to 9, the dialer 466, transaction server 468, and roam server 472 include an ECC API that implements the ECC algorithm and provides an API for encrypting and decrypting passwords. Typically, the ECC implementation uses optimal normal basis mathematics for encryption/decryption. In certain embodiments, polynomial basis and optimal normal basis mathematics are combined to reduce the time for a mathematical inversion to the cost of a single multiply.

[00096] Referring in particular to Figure 10, reference numeral 580 generally indicates exemplary encryption functionality of the dialer 466. As shown at block 582, the encryption algorithm generates a random point on an ECC curve. This random point is then used for encoding the password and the checksum character (see block 584) to produce part of an ECC string <encoded password>. The dialer 466 encrypts the random point and transmits it to the netserver 470 (see blocks 586 and 587). Typically, a symbol transformation scheme is used for this encryption as described below.

[00097] In order to accommodate existing protocols, e.g., PPP, PAP, RADIUS, or the like, the password fields have printable US-ASCII characters. In certain embodiments, the characters are generated in such a fashion so as to conform to RFC 2486 standards. In these embodiments, when the password and checksum fields are encrypted, care is taken to generate the string with acceptable characters so that they may be applied in networks using standard protocols (see block 588). Accordingly, the following character transformation scheme may be used to perform this encoding. Each character to be encoded is first mapped into a value according to the table shown below.

#	SYMBOL	#	SYMBOL	#	SYMBOL	#	SYMBOL

#	SYMBOL	#	SYMBOL	#	SYMBOL	#	SYMBOL
0.	0	1.	1	2.	2	3.	3
4.	4	5.	5	6.	6	7.	7
8.	8	9.	9	10.	A	11.	B
12.	C	13.	D	14.	E	15.	F
16.	G	17.	H	18.	I	19.	J
20.	K	21.	L	22.	M	23.	N
24.	O	25.	P	26.	Q	27.	R
28.	S	29.	T	30.	U	31.	V
32.	W	33.	X	34.	Y	35.	Z
36.	A	37.	B	38.	C	39.	D
40.	E	41.	F	42.	G	43.	H
44.	I	45.	J	46.	K	47.	L
48.	M	49.	N	50.	O	51.	P
52.	Q	53.	R	54.	S	55.	T
56.	U	57.	V	58.	W	59.	X
60.	Y	61.	Z	62.	~ (TILDE)	63.	` (GRAVE ACCENT)
64.	! (EXCLAMATION MARK)	65.	# (NUMBER SIGN)	66.	$ (DOLLAR SIGN)	67.	% (PERCENT SIGN)
68.	^ (CARET)	69.	& (AMPERSAND)	70.	* (STAR SIGN)	71.	((LEFT PARENTHESIS)
72.	) (RIGHT PARENTHESIS)	73.	- (HYPHEN-MINUS)	74.	_ (UNDERSCORE)	75.	+ (PLUS SIGN)
76.	= (EQUALS SIGN)	77.	{ (LEFT CURLY BRACKET)	78.	[(LEFT SQUARE BRACKET)	79.	} (RIGHT CURLY BRACKET)
80.	] (RIGHT SQUARE BRACKET)	81.	| (VERTICAL LINE)	82.	\ (REVERSE SOLIDUS)	83.	: (COLON)
84.	; (SEMICOLON)	85.	" (QUOTATION	86.	' (APOSTROPHE)	87.	< (LESS-THAN

#	SYMBOL	#	SYMBOL	#	SYMBOL	#	SYMBOL
			MARK)				SIGN)
88.	, (COMMA)	89.	> (GREATER-THAN SIGN)	90.	? (QUESTION MARK)	91.	(SPACE)
92.	/ (SOLIDUS)	93.	. (FULL STOP)	94.	@ (COMMERCIAL AT)		

[00098] The mapped value is then added to the corresponding byte in the random point and the modulus 95 is calculated (see block 590). This results in the character being mapped to another character in the above table. To decode the character at a decryption server, the corresponding byte in the random point is subtracted from the encoded character (see block 581 in Figure 11) and the modulus 95 of the result is calculated (see block 583). If the result is a negative number, then the value 95 is added to the result to obtain the original character (see block 585). By way of illustration, assuming "r" is the byte in the random point used for the encoding, and "x" is the original character, then,

Encode: $y = (x+r)\%95$

Decode: $x = (y-r)\%95$

If $(x < 0)$ then

$x = x+95;$

[00099] The password field and the checksum character are encrypted with the random point during the encryption process at the dialer 466. Each one of these fields uses a different set of bytes in the random point for encoding. The password field uses the first set of bytes for it's encoding, and the checksum field uses byte 10 for it's encoding.

[000100] The checksum character is used for ascertaining the integrity of the dialer id and counter values. If the dialer id and the counter value are transmitted in the clear, a malicious person can alter these values and thereby defeat the protection against replay attacks. To address this problem, a checksum character is generated from the dialer id and counter value where after it is encoded using the random point (see block 592 in Figure 12). The encrypted checksum character is then transmitted as part of the user id string.

[000101] The checksum character is generated by the MD5 hash of the count value, the dialer id and the random point (see blocks 592 and 594 of Figure 12). Seven bits are then selected from the hash and then encoded with a single byte (byte #10) from the random point (see block 596) using the encoding methodology described above. The encoded bits are then dispersed among the last seven bytes of the encypted point (see block 598) and transmitted as part of the user string (see block 599). When the dialer 466 sends the encoded data to the transaction server 468 or roam server 472, as the case may be, they validate the dialer id and counter value by independently generating the checksum (see block 588 in Figures 8 and 9) and compare it with the checksum sent by the dialer 466 (see block590) and reject if they don't match.

[000102] Returning to the dialer 466 and to Figure 10, the encoded strings are then concatenated as follows to create an ECC string:

<encoded password><encrypted and encoded x coordinate of the random point with encoded checksum bits in the last seven bytes>

[000103] Thereafter, the dialer 466 concatenates the ECC string with the dialer id and the counter value and transmits it in the userid and password fields of the protocol, e.g. PAP. For example, <encoded password><encrypted and encoded x coordinate of the random point with encoded checksum bits in the last seven bytes><dialer id><counter value> .

[000104] It will be noted that the methodology set out in Figure 10 produces an encrypted string that is of such a string length, and includes characters of such a nature, that the encrypted string may be communicated using LEGACY systems.

[000105] The encryption logic is typically encapsulated in an ip_spap_encrypt()method with the following signature:

char * ip_spap_encrypt(const char *algorithm, const char public_key, const char password, const char *dialer_id, const char *counter, char **plain_point, char **encrypted_point, int *returnCode);

where

algorithm is the algorithm to be used. "S" for Secure PAP

public_key is the ECC public key (from config.ini)

password is the plain-text password

dialer_id is the id of the dialer (obtained from the dialer id servlet)

counter is the count of dial attempts (incremented by the dialer for each dial attempt)

plain_point - If this field is left empty, a new random point is generated. This field points to the random point used for the encoding on return.

encrypted_point - If this field is left empty, the plain point and the public key is used to generate the encrypted point. This field points to the encrypted point used by the method on return.

returnCode 0 if the call is successful, a non-zero code is provided. The method returns the ECC string is returned when successful and a null otherwise.

[000106] The decryption logic is encapsulated in the ip_spap_decrypt()method. The method have the following signature:

char * ip_spap_decrypt(const char *algorithm, const char private_key, const char ecc_string, const char *dialer_id, const char *counter, int *returnCode); where

algorithm is the algorithm to be used. "S" for secure pap

private_key is the ECC private key (from securepap table or esp_key_pair.txt file)

ecc_string is the string returned by the encrypt() method

dialer_id is the id of the dialer (obtained from the dialer id servlet)

counter is the count of dial attempts (incremented by the dialer for each dial attempt)

returnCode 0 if the call was successful; non-zero code otherwise

[000107] The method returns the plain text password when successful and a null otherwise.

Dialer Customization Form

[000108] As mentioned above, the customers 456 use a web form for requesting a customized dialer configured to communicate using Secure PAP. This web form typically contains fields that can be used for specifying the

required customizations. The web form may include the following exemplary fields:

Enable Secure PAP encryption: (Y/N)

Public Key:

Key Id: (0-9)

Dialer Customization Tool

[000109] During the customization process, an administrator of the access broker system 454 has the option of generating a dialer 466 that will use Secure PAP. If enabled, the following exemplary fields may be set in a config.ini that is typically packaged with the dialer 466:

[processing facility identification e.g., iPass]

EncryptFlag=yes

Algorithm=S

KeyVersion=0

PublicKey=BwAAAMGdqYx2lxhWtEQMdDHhvwU=&AQAAAFdd 40uLQMD1UTtyBqDHY=

[000110] These values are also stored in the transaction server database so that the transaction server 468 can decrypt the password sent from the corresponding dialer 466 of a particular customer 456. In the present embodiment, only the public key is stored in this file, as the private key is kept secret for the encryption to be secure.

[000111] In addition to enabling Secure PAP, the customization tool also provides the option of setting the algorithm used and the key version. For example, the following encryption algorithms may be supported:

A for no encryption.

E for Elliptic Curve Encryption

S for ECC compatible with Unique Session ID

U for Unique Session ID

[000112] In practice, A is primarily for testing and debugging purposes. E is used for encrypting the password when the dialer does not have the dialer id. U is not an encryption algorithm, but is used to identify the unique session id, as discussed in more detail below. The key version starts at zero, but is

incremented every time a new key-pair is desired for an existing dialer profile. The dialer 466 stores the ECC keys and other information in a secure_pap table. This table is then replicated to the transaction server 468 via Oracle snapshots. A new key-pair is generated if the private key has been compromised. When the security of the private key is compromised, the dialer support team should take the following actions:

1. Set an appropriate expiry date for the compromised key. This should be sufficient to ensure all dialers 466 using the compromised key can still use the key one last time. The dialers 466 connect to the Internet using the old key, and retrieve the config.ini file with the new key from the update server. If the customer 456 is using the roam server 472 to decrypt the password, the customer 456 typically manually removes the compromised key from the esp_key_pair.txt file after the expiry date.

2. Generate a new key pair or ask the customer 456 to generate a new key pair and send the public key to the access broker system 454.

3. Use the DCT tool and replace the public key (use a new key id). Build the dialer.

Dialer

[000113] The dialer 466 checks the config.ini file to determine whether or not it should be encrypting passwords. If Secure PAP is enabled, then the dialer 466 encrypts the password using the public key from the config.ini file and by invoking the ip_ spap_encrypt() method. The method creates the ECC string and returns it. The dialer 466 concatenates the ECC string with the dialer id and the counter value. The first sixteen characters of the ECC string are placed in the password field and the rest of the string is placed in the prefix field (with 0S or 0E prefix). The dialer 466 uses algorithm "E" until it obtains a dialer id. The prefix is included after all system and routing prefixes, but before the customer prefixes. The dialer 466 does not encrypt the password and does not create the Secure PAP prefix if the POP being dialed has a prefix that is not compatible with and PAP prefix in the phonebook. A sample username, which includes the encryption prefix is a follows:

UserID: IPASS/0S Axri50zTxca546hjdgkbxcjc^_d0we/joe@ipass.com

Password: x35~!4Qu{xy71]D8

where KeyVersion=0 and Algorithm=S.

[000114] If the access broker system 454 determines that no encryption is needed, it creates a unique session id from the dialer count and places it in the prefix field. A sample username, which includes the unique session id prefix is as follows:

UserID: IPASS/0UAxrt5AB2/joe@ipass.com

Password: thisisabigsecret

where KeyVersion=0 and Algorithm=U.

The dialer 466 stores the plain_point and the encrypted point in its local storage.

[000115] When a redial is attempted, the dialer 466 increments the counter and invokes the ip_spsp_encrp() method using the plain point, and encrypted point.

Customer Resolution

[000116] The customer resolution process checks for a prefix of the form [0-9][A-Z]*/ If no such prefix is found, and the customer 456 does not require password decryption, the customer resolution operates as normal. If the prefix is found, the last 8 bytes up to the first slash (/) are stripped out and stored as the unique session id field. The customer resolution code may create the unique session id field with the following contents: 05<dialer_id><counter>. The integer is stripped and stored as key identifier field. The algorithm is stripped and stored as a separate field.

Dialer Counter Replication

[000117] Secure PAP embodiment depicted in the drawings uses the dialer_counter table for protection against replay attacks. Each transaction server database contains a dialer_counter_ts table. The transaction server 468 inserts a new row into this table whenever it receives a successful authentication request with a Secure PAP prefix. The contents of this row include the server_id, the dialer_id, the counter and the system time (in GMT).

[000118] The SESSION database contains a snapshot for the dialer_counter_ts table at every transaction server 468. These snapshots are typically named:

dialer_counter_ts_<SERVER_ID>, where <SERVER_ID> is the id of the particular transaction server 468.

[000119] A "refresh" tool is provided for refreshing the snapshots from the transaction servers 468. The dialer_counter_ts_<SERVER_ID> would have "ON INSERT" PL/SQL trigger that would update/insert the dialer_id, counter, and access_time from the inserted row into the dialer_counter table if the value of the counter being inserted is equal to or greater than the value of the counter in the dialer_counter table. The transaction servers 468 use the refresh tool to refresh the dialer_counter snapshot from the SESSION database. The dialer_couter table is then cached by the transaction servers 468 for faster access. Any changes to records in dialer_counter table at runtime take immediate effect. This is accomplished using the same mechanism used in other components of the access broker system 454 using database triggers and the cache_update table.

Transaction Server

[000120] On startup, the access broker system 454 reads all private keys from the database into a local cache for efficient lookup. It also has an additional attribute in the customer cache to indicate if a certain customer 456 requires password encryption or not. The transaction server 468 also caches the dialer_counter table. Any changes to records in these tables at runtime take immediate effect. This is accomplished using the same mechanism used in other components of the access broker system 454 using database triggers and the cache_update table.

[000121] If the encrypted prefix field specifies the 'S' algorithm, the transaction server 468 concatenates the contents of the password field to the encrypted prefix field constructed by the customer resolution process and creates the "ECC field". The ECC field contains

<encoded password><encrypted and encoded x coordinate of the random point><encoded checksum character>

[000122] The transaction server 468 locates the private key for the appropriate customer 456 using the key index. If the private key is found in the database, it calls the ip_spap_decrypt() method to decrypt and decode the password. The

password field is then overwritten with the plain-text password before it is sent to the roam server 472.

[000123] If the private key is not located in the cache, the transaction server 468 typically adds the following fields to the authentication request packet and sends it to the roam server 472: algorithm, key index, the ECC field (as password), dialer id, counter, value and access time of the counter last used (from the database), and the "decrypt_at_roamServer" flag set to "yes".

[000124] The transaction server 468 then stores the authentication details in the ip_auth_trans table and the dialer_counter details in the dialer_counter_ts table. The Transaction server 468 typically inserts a new dialer_counter_ts record every time as inserts are usually faster than updates.

[000125] When the transaction server 468 receives the account request, it uses the customer resolution process to create the unique session id and adds it to the packet as "ipass_session_id". The tr_userid field contains the raw_userid.

ESP Tool

[000126] The ESP command line tools are used by the customers 456 in conjunction with their roam servers 472, the DCT team, and the QA team to generate public/private key pairs and test the encryption and decryption algorithms.

esp_genkey (for customers 456 with roam servers 472):

[000127] This tool prints the public/private ESP key pair to a file named *esp_key_pair.txt*. This file resides in the */usr/ipass/keys* directory on Unix, and in the *IPASS_HOME/keys* directory for Windows. The keys must also be submitted to the access broker system 454 via, for example, a secure website so that the dialer 466 can be built with the public key. Typically, a secure backup of the private key is also maintained.

esp_genkey_dct:

[000128] This tool prints the public/private ESP key pair to standard output. It is printed in a format that meet the requirements of the DCT. An example output is:

1

Public Key:BgAYVK1azUt8comk41GzLw=&ADIkGfMgNChM4vY6+n LgTqo=

Private Key:AQAAAAZOSNH13PaG3NuqGbU7TY0=

[000129] The first line contains a "1", indicating success in key generation. When an error occurs, that output is then of value "0".

esp_qa:

[000130] This tool has several command line options available for testing the ECC API. An example sample of the option supported:

esp_qa genkey

esp_qa encrypt [-a <algorithm> -d <dialer_id> -c <counter>] -k <public_key> -t <text>

esp_qa decrypt [-a <algorithm> -d <dialer_id> -c <counter>] -k <private_key> -t <text>

esp_qa testipg [-a <algorithm> -d <dialer_id> -c <counter>] -k <public_key> -t <text> -u <uid

@domain>

esp_qa test -t <text>

Option in brackets[] are optional. Each esp_qa command are described as follows:

genkey - Generate a public/private key pair.

encrypt - Encrypt text (password) with the given public_key.

decrypt - Decrypt text (password) with the given private key.

testipg - Executes the "Encrypt" then runs the check-ipen command for the given user.

test - Basic ECC API test. Runs the genkey, encrypt, and decrypt for algorithm S.

Roam server

[000131] The roam server 472 typically checks for the presence of the "decrypt_at_roamserver" field in the packet received from the transaction server 468. If the field is present, the roam server uses the "key index" field from the packet and fetches the private key from the esp_key_pair.txt file. The ECC string along with the private key, dialer id and counter value is passed to

ip_spap_decrypt() method. The ip_spap_decrypt() method decodes and decrypts the password. The plain text password is then used by the roam server 472 to authenticate the user.

[000132] Returning to Figure 6, once the dialer 466 has performed the methodology set out above, the authentication data is communicated to the NAS 532 where after it is sent to an authentication server 600 of the remote ISP 506. In the normal course of operations, the NAS 532 at the remote ISP 506 would reject the supplied authentication information. However, as illustrated in Figure 6, the netserver 470 intercepts the authentication information to facilitate recognition of this authentication information as a roaming user authentication request and not a regular user request.

[000133] The authentication server 600, in conjunction with the netserver 470, parses the received authentication information to determine a roaming domain name or routing prefix associated with the roaming user 502. Should such a domain name or prefix be present, the user's authentication information is encrypted as set out above, and sent from the netserver 470 to the transaction server 468 via a secure socket layer (SSL).

[000134] The transaction server 468 may use a customer routing prefix in the session identification to route the request. Instead, the transaction server 468 may perform an Internet Protocol (IP) look-up and routes the authentication request to an appropriate home ISP 504. More specifically, the transaction server 468 receives the encrypted authentication request from the netserver 470 at the remote ISP 502, and decrypts this request as described above with reference to Figures 7 to 9. The transaction server 468 then determines the "home" ISP 504 by matching the roaming domain name or routing prefix of the desired home ISP 504 against a current list of participant domain names and IP addresses. If the match is successful, the authentication request is encrypted and sent via SSL to the roam server 472 that resides at the home ISP 504. In the event that the identified roam server 472 does not respond within a specific period, the transaction server 468 will attempt to contact an alternative roam server 472 at the ISP of the relevant domain.

[000135] The roam server 472 at the home ISP 504 then decrypts the authentication request sent from the transaction server 468, as described above, and submits the authentication request to the home ISP's regular authentication server 602 as if it were a terminal server or NAS 532 owned by the home ISP 504 using a customer prefix. The authentication server 602 of the home ISP 504 responds to the request by providing an "access permitted" or an "access denied" response based on the validity of the user name and password included within the authentication request (see Figure 8). The response from the home ISP's authentication server 602 is received by the roam server 472, encrypted, and sent back to the transaction server 468.

Unique session identification

[000136] An exemplary method and system to associate a plurality of transaction data records is described below. The method and system describes the generation and use of a unique session id which is typically used in combination with the encryption/decryption methodology described above.

[000137] As mentioned above, communication protocols such as, for example, Point-to-Point protocol (PPP), Password Authentication Protocol (PAP), Challenge-Handshake Authentication Protocol (CHAP), Remote Authentication Dial In User Service (RADIUS) protocol, Terminal Access Controller Access Control System (TACACS) protocol, Lightweight Directory Access Protocol (LDAP), NT Domain authentication protocol, Unix password authentication protocol, HyperText Transfer Protocol (HTTP), HyperText Transfer Protocol over Secure sockets layer (HTTPS), Extended Authentication Protocol (EAP), Transport Layer Security (TLS) protocol, Token Ring protocol and Secure Remote Password protocol (SRP) make provision for a user identification string. Although the size or length of characters that each different protocol allows may vary, the lowest common denominator in size supported by the exemplary protocols listed above is typically about 63 characters. In these circumstances, provision of a unique user session identification would enhance authentication, accounting and SQM processing.

[000138] In the application of above protocols to the exemplary multi-party service access environment 450, the user identification string is included, and is

thus common, in all relevant transactions data records generated by the various participants, such as the transaction servers 468, the service providers 452 and the customers 456. However, in certain circumstances, although the prior art user identification string used in these protocols may be uniquely associated with a particular user of multi-party service access environment 450, it is not uniquely associated with a particular single user session. For example, due to network timeouts and packet retry algorithms, it is often the case that a single transaction data record is sent to a transaction servers 468 several times and, if any one or more of these records is defective, multiple instances of a record relating to the same single user session may exist at the settlement system 474. Further, in an attempt to re-send a perceived failed communication attempt, certain NASs 470 (see Figure 6) actually change the user session identification string thereby resulting in different transaction data records for the same single user session. The aforementioned are merely two examples of unsatisfactory accounting records but it will be appreciated that there may be a host of other circumstances.

[000139] In accordance with another embodiment of the present invention, relevant transaction data records generated in response to a single user session include a common unique session identification. In certain circumstances, this session identification may provide strong, but not necessarily absolute identification of an individual user's usage information and the unique user session identification should at least be unique within certain parameters. For example, the unique user identification my be unique for a given time period so that all records generated during that time period may be associated and processed using the unique user session identification.

[000140] Typically, for the exemplary protocols mentioned above, the user identification string includes, not only a user name and password of the user accessing the network, but also routing information including the customer realm. The user ID or identification string used in the exemplary multi-party service access environment 450 is typically as follows:

<FacilityRoutingPrefix>/[<FacilityLocationPrefix>]/[<CustomerRoutingPrefix>]/[CustomerPrefix(s)]/<EndUserName>@[<NonRoutingCustomerDomain>] |[<CustomerRoutingDomain>]

Wherein,

<FaclityRoutingPrefix> is a proprietary prefix that is used by the ISPs 458, wireless access providers 460, content distribution provider 462, E-Commerce provider 464, or the like (the access providers) to route traffic to the network of the access broker system or facility 454.

<FacilityLocationPrefix> is a prefix used by the facility to determine the location of points or nodes providing access to the facility 454.

<CustomerRoutingPrefix> is a prefix used by the access or service providers 452 to route traffic to the customer site.

<CustomerPrefix(s)> is a/are prefix(es) used by the customer 456 for their internal routing.

<EndUserName> is the login user name of the end user 502 using the facility 454.

<CustomerRoutingDomain> is a domain used by the system 454 to route traffic to the customer site. The user ID string includes either the <CustomerRoutingPrefix> or the <CustomerRoutingDomain>.

<NonRoutingCustomerDomain> is a domain used by the customer 456 for their internal routing.

[000141] An example of one of the possible ways of fitting the unique session identification in the user identification field of one of the above protocols is now described. It will however be appreciated that the inclusion of the unique session identification may be implemented in other ways. An example of an alternative solution is implemented when the dialer uses the E type algorithm for password encryption. The E type algorithm includes the encrypted random point in the username. The encrypted random point provides strong, but not necessarily absolute identification of the individual users session, and so is used as the unique session id.

[000142] As mentioned above, the lowest common denominator available string length for proprietary information supported by the exemplary protocols

is typically about sixty-three characters. The unique session id should fit within the limits imposed by the username field.

[000143] In order to generate the unique session identification (see block 802 in Figure 20), the connection application 466, in the exemplary form of a connect dialer, resident at each service provider 452, obtains a dialer identification which identifies the connect application 466 from a servlet in the web server 806 of the access broker system 454 (see Figure 15). The dialer identification is typically also a unique dialer identification. The dialer identification is stored in a user preference file and, when the dialer is initially installed; the dialer identification in the user preference file is typically empty. The first time the dialer 466 connects, for example, to the Internet, it typically requests a new dialer identification from the web server 806 (see block 800). In the embodiment shown in the drawings, the dialer does not create a unique session identification until it obtains the unique dialer identification from the web server 806. Accordingly, in this embodiment where the dialer identification forms part of the unique session identification, the first successful session from the dialer 466 would not contain a unique session identification. The dialer 466 would however have its dialer identification for any subsequent attempts.

[000144] In addition to its own dialer identification, the dialer 466 also includes a counter 467 that is internally maintained and stored in the user preference file. The counter 467 is incremented for each dial attempt (see block 802). The dialer 466 using its dialer identification and the counter generates a session identification indicator, defined by eleven characters (see Figure 18) in the present embodiment, at each subsequent dial attempt. As the counter 467 is incremented at each dial attempt, the dialer generates a globally unique session identification: <dialer id><counter>(see block 802). In this embodiment, the session identification is prefixed by an identifier, e.g., three characters such as "OU" associated with the facility or access broker system 454, which are stripped off by the transaction server 468 before the user identification string is passed onto the roaming server 472 (see Figure 5). Thus, when the unique session identification includes eleven characters, eight characters would be available for the dialer identification and counter.

[000145] Both the exemplary dialer identification and counter use numbers with radix 64. The symbols used for this numbering scheme include A-Z, a-z, 0-9, & and ^. The counter 467 is incremented prior to each dial attempt and the dialer identification is pre-filled with zeroes and, in the present embodiment, defined by a five digit entry. Accordingly, three digits remain for the counter 467. Accordingly, the five digits used for the dialer identification would enable 1073741824 unique dialer installs (more than a billion) and the three digit counter enables 262144 dial attempts (the counter would reset after 23 years, assuming 20 attempts a day). During this period, the session identification would thus uniquely define each user session. It is however to be appreciated that the number of characters allocated or used for the unique session identification may vary from system to system dependent upon the type or types of protocols that the system accommodates.

Transaction Record Processing

[000146] Figure 14 is a block diagram illustrating the accounting and settlement procedures, according to an exemplary embodiment of the present invention, which may be facilitated by the access broker system 454.

[000147] When a roaming user 502 connects to the remote ISP 506, the terminal server (or NAS) 470 managing the session generates a transaction data record that includes the user identification string, and thus the eleven character unique session identification, and sends this information to the authorization server 600. The authorization server 600, in conjunction with the netserver 470, parses the accounting information to determine a roaming domain name and prefix associated with the roaming user. Should such a domain name or prefix be present, the user's accounting information is encrypted using an algorithm from RSA Data Securities, and sent from the netserver 470 to a transaction server 468 via secure socket layer (SSL).

[000148] When a roaming user 502 disconnects from remote ISP 506, the terminal server (or NAS) 470 managing the session generates a transaction data record that includes the user identification string, and thus the eleven character unique session identification, and sends this information to the authorization server 600. The authorization server 600, in conjunction with the netserver 470,

parses the accounting information to determine a roaming domain name and prefix associated with the roaming user. Should such a domain name or prefix be present, the user's accounting information is encrypted using an algorithm from RSA Data Securities, and sent from the netserver 470 to a transaction server 468 via secure socket layer (SSL).

[000149] A transaction data or accounting record is then communicated, in near real-time, to the transaction server 468 utilizing SSL, where the accounting records are stored in the database. All the various components or participants in the multi-party service access environment 450 receive the user identification string, and thus the unique session identification, which then accompanies the transaction data record associated with the single user session when the transaction data record is sent to the settlement system 476. Thus, transaction data records sent from various different participants include an identifier that identifies the single user session from which they arise.

[000150] These accounting records are further processed by the settlement system 476 to produce Call Detail Records (CDRs). Each call detail record provides detailed usage reporting regarding the identity of the roaming user 502, when the relevant service access occurred, the location of the service access, the length and cost of each service access session, and the time of the service access (e.g., local or GMT time).

[000151] Multiple transaction servers 468 provide accounting or transaction data records to the settlement system 476, which utilizes these records to generate bills (or invoices) to customers 456, and also to make payments to service providers 504. It is, however, to be appreciated that accounting information sent to the transaction server 468 may, for various reasons, be incomplete, differ from one ISP to the next, be sent more than once and so on. Thus, a variety of different, and possibly incomplete, records relating to the same single user session may be received by the transaction server 468.

[000152] Naturally, identifying or associating all transaction data records arising from a particular user session is advantageous in that the settlement system 474 generates bills and distributes them among customers 456 so that they can make payments to the settlement system 474, and in turn bill their

customers if appropriate. Similarly, the settlement system 474 makes payments to the remote (or visitor) ISPs or other service providers 452 for accrued access time used by roaming users. The settlement system 474 may further guarantee payment for authorized use by a roaming user. An operator of the settlement system 476 thus acts as a secure, trusted entity providing a mechanism for facilitating financial settlement of service access transactions between multiple parties. The settlement system 476 implements numerous automatic functions and operations so as to enable the settlement in a timely, automated and convenient manner. Further details regarding the operation of the settlement system to facilitate such settlement or service access transactions will be described in detail below.

Physical Architecture

[000153] Figure 15 is a diagrammatic representation of the physical architecture of the access broker system 454, according to an exemplary embodiment of the present invention. Multiple transaction servers 468 are shown to reside on one or more server machines 810, each of which has access to an associated database 812. A web server and phonebook server reside on the server machine 806, and are accessible by remote internal users 814 and the customers 456. The web server operates to generate and deliver web pages (e.g., HTML documents) to both the remote internal users 814 and the customers 456. As described above, in one embodiment of the invention, a servlet on the web server residing on machine 806 provides a unique connection application identification, in the exemplary form of a dialer identification, to each dialer or connection application 466 residing with the services providers 452. The phonebook server (part of the phonebook management system 480) operates to maintain and update the electronic phonebooks of customers 452, and accordingly both receives and publishes updates to and from service providers 452, and publishes such updates to customers 456.

[000154] The settlement system 476, and a collection of internal users 816 are shown to reside behind a firewall 818. Specifically, the settlement system 474 is

hosted on one or more server machines 820 that have access to a central database 822.

Overview - Settlement System

[000155] Figure 16 is a block diagram illustrating the architecture of a settlement system 474, according to an exemplary embodiment of the present invention. The settlement system 474 comprises a back-end applications 824, front-end applications 826, data aggregation and reporting applications 828 and system interfaces 830.

[000156] The back-end (or server-side) applications 824 are shown to include a settlement application 832 that determines a transaction price, updates account balances for all parties involved in a transaction, and verifies credit limits, a billing application 834 that closes an accounting cycle, applies periodical fees, generates billing reports, including invoices and call detail records (CDRs), and publishes billing reports to the web, and an auditing application 836 that verifies business rules and structural integrity of the central database 822. The settlement application 832 is shown to embody the flexible pricing engine 476.

[000157] In the present embodiment, the settlement application 832 is responsible for normalization, summarization and verification functions. The normalization function includes converting accounting data received from multiple transaction servers 468 into a single format CDR to be used for billing, identifying parties involved in a service access transaction, and defining the price that the access broker system 454 owes to a provider 452 and the price that a customer 456 owes to the access broker system 454 for a particular service access transaction. The summarization function involves applying buy and sell prices to account balances for all parties involved in a service access transaction, and updating appropriate account balances. The verification function includes the verification of credit limits.

[000158] The settlement system 474 operates to provide near real-time settlement of service access transactions to allow for the near real-time revenue and account tracking by both providers 452 and customers 456.

[000159] In certain embodiments of the invention, the settlement system 474 includes the flexible pricing engine 476 that supports a flexible pricing model, which has the following features:

1. A variety of data structures dependent on, for example, the customer 456, the service provider 452, the location of the service access, the type of service access (e.g., dialup modem, ISDN, DSL), or usage accumulated during a particular cycle for a particular customer 456.

2. Any combination of (a) usage (e.g., a function of rate and session length); (b) transactional (per transaction); and (c) subscription-based or flat pricing (e.g., one price for all usage during a billing cycle for a customer 456 or one or more prices per each user for a customer during a billing cycle).

3. Offered discounts and promotions.

4. A variety of fees, such as start-up fees, monthly fees and minimum monthly commitments.

5. Multi-tiered pricing schemes, or intra-provider roaming, where buy and sell rates for a particular location depend on the provider 452 and whether the service user/customer 456 of the service access belongs to a further customer 456, its affiliate, or their customer.

[000160] The flexible pricing engine 476 is database-driven, thus allowing implementation of new pricing models by loading the appropriate plan into pricing tables (not shown) maintained within the central database 822. More specifically, the flexible pricing engine 476 facilitates a multi-tiered pricing model, whereby rates for a single service access transaction may be applied across multiple tiers of consumer (or customer) according to multiple criteria. These criteria may include, *inter alia*, any combination of usage (e.g., accumulated usage time or value total) pricing and transactional (e.g., an accumulated total number of transactions) pricing.

[000161] Returning now to Figure 16 and the front-end applications 826, a data management application 838 is utilized by various functional units of the access broker to perform business processes and to view data for information purposes. To this end, data management application 838 may provide various

user interfaces to manage information related to customers 456 and access points, and to perform accounting and administrative functions.

[000162] An order processing application 840 provides user interfaces to customers 456 (e.g., solution partners 488 or resellers) to place orders for new corporate customers.

[000163] The data aggregation and reporting applications 828 include several processes that summarize data on a daily or monthly basis to enable operational, functional and network load reporting.

[000164] The system interfaces 830 have a loader application that includes a transaction server loader 842, a provider loader 844 and accounting system interfaces (not shown). Dealing first with the transaction server loader 842, a "data loader" component pulls accounting records in the form of transaction data records, including the unique session identification, from the databases 812 of the respective transaction servers 468 to the central database 822 for processing. Multiple transaction server or batch loaders 842 may be implemented as distributed database links, and the accounting or transaction data records are pulled via the loaders 842 in near real-time.

Overview - Data Model

[000165] Figure 17A is a block diagram illustrating an exemplary data model 845 including customer tables 846, access point tables 848, pricing tables 850, CDR tables 852, accounting tables 854, authentication transaction storage area or tables 856, batch history storage area or tables 858, and SQM storage area or tables 860.

[000166] The network components in the access broker system 454 may, in certain embodiments, strip the routing prefixes from the transaction data records. Some of these components may also truncate the user identification string. The Unique session id prefix is neither a routing prefix nor at the end of the username, hence it is neither stripped nor truncated The user identification string is thus processed to remove these defects before it is used to uniquely define the user session. A modified user session identification is constructed using as many of the following components that are available:

<AuthCustomerId>/<UniqueID>/[CustomerPrefix(s)]/<EndUserName>@<NonRoutingCustomer Domain>

Wherein,

<AuthCustomerId> is the authenticating customer identification, produced by the customer resolution process.

<UniqueID> is the unique session identification code, 0Uxxxxxxx/, prefix generated by the connection application 466 as described above.

<CustomerPrefixe(s)> are prefixes used by the customer for their internal routing as described above.

<End User Name> is the user identification of the end user connecting to the access broker system 454 as described above.

<NonRoutingCustomerDomain> is the domain used by the customer for internal routing, as described above.

[000167] Referring in particular to Figure 17B, provider loader 844 receives call detail records (CDRs) or transaction data records, including the unique session identification, from the providers 452 in a batch form. This CDR data is pre-processed by the provider loader 844, which may retrieve the data from an appropriate FTP site and convert it into the same format as the data received from the transaction servers 468. In particular, the transaction server 468 constructs, each time a user access session is authorized as described above, a modified user session identification and stores it a session_id field in authentication transactions tables 856 and a session_id field in account transaction tables 854 (see Figure 17A). It will be appreciated that, for each modified session identification stored in the authentication transactions table 856, corresponding transaction data records should be received by the settlement system 474 for processing. In a similar fashion to the transaction server 468, the batch loaders 842, 844 respectively construct or build a modified transaction data record from each transaction data record received from the transaction servers 468 of the service providers 452 (see Figure 5 and 17B). The modified session identification from the loaders 842, 844 are stored it in a session_id field in the batch history tables. Likewise, the SQM process

constructs the modified session identification and store it in a session_id field in an SQM table 860.

[000168] The use of the unique session identification including its unique code may be used in addressing the following issues.

Missing Accounting records

[000169] Missing accounting/transaction data records (see block 862 in Figure 19) may arise for various reasons such as delivery failure, malformed records, misrouted records, or the like. Delivery failure may occur when the Internet connectivity from the ISPs (e.g., the netserver 470 in Figure 6) is disrupted, thereby blocking the delivery of the transaction data record to the settlement system 476. A connectivity outage that persists for more than a few minutes typically causes the netserver 476 to discard the transaction data record due to minimal transmission retry capabilities. When using the RADIUS protocol, malformed records are typically discarded at any of several intermediate points including the authentication server of the service provider (e.g., the authentication/authorization server 602 or netserver 470 (see Figure 6)). Misrouted records are records not sent due to an improper configuration of the ISPs authentication server 602 either accidentally or with fraudulent intent.

[000170] As every access session by the user first requires authorization, each session_id field in the authentication transaction tables 856 should include a corresponding session_id field in the acct_trans table. Accordingly, by associating, matching, correlating, investigating, the session_id fields, missing accounting/transaction data records can be determined. In the embodiment depicted in the drawings, missing accounting/transaction data records typically would have an authentication request record in the authentication transaction or auth_trans table 856, but no matching accounting start and/or accounting stop records in the accounting or acct_trans table 854. Thus, by searching for all session_id fields in the acct_trans table that correspond to each session_id field in the auth_trans table, missing accounting/transaction data records may be found (see blocks 864-872).

Inappropriate accounting records

[000171] Inappropriate accounting/transaction data records may be received by the settlement system 474 (see Figure 6) usually due to inappropriate configuration of a provider's authentication server (AAA server), e.g. server 600 in Figure 6. An inappropriate configuration typically causes the provider's authentication server 600 to send all accounting/transaction data records to all proxies instead of just the one responsible for the authentication of the user access session. In these circumstances, no session_id field is present in the auth_trans table of an incorrect recipient. As these accounting/transaction data records typically do not have a corresponding authentication record, they may be identified with relative ease and, for example, a customer support team can resolve the configuration problem with the provider and prevent recurrence of such incorrect transmissions. In these circumstances, the methodology shown in Figure 19 may be used except that the auth_trans table is searched for a unique session identification corresponding to an entry in the acct_trans table.

Duplicate Accounting records

[000172] Duplicate accounting records are multiple transaction data records that describe the same single user access session. In the embodiment depicted in the drawings, duplicate transaction data records are actively filtered by the settlement system 474 using a relatively simple algorithm that matches six "key" fields of each real-time accounting/transaction data record against all other real-time accounting/transaction data records that have been received within the previous 30 days. The exemplary fields used are: RADIUS Session-Id, Provider ID, NAS IP Address, User, Domain (user auth realm), and Session Time.

[000173] In certain embodiments, when all six fields match those in an already-rated record in the CDR table, the current record is marked as a duplicate and discarded.

[000174] Duplicate accounting records may arise for a variety of reasons:

[000175] Accounting/transaction data records must be acknowledged through the timely transmission of an Accounting-Response message to the sender. Unfortunately "timely" is not defined by the RADIUS specification and

different vendors and configurations may resend unacknowledged accounting transactions a few seconds, hours or even days later. When the accounting request was actually received by the settlement system but the acknowledgement was lost or malformed, the originator may resend multiple copies of the accounting record. All such records are captured by the receiving transaction server 468 and eventually retrieved by the settlement system 474 for processing. Peculiar variations on this class may occur which elude the settlement duplicate filtering algorithm wherein the sending NAS 532 sends an updated (e.g., incrementally longer) session time with each retransmission or the RADIUS Session-Id changes between retransmissions. In some cases the NAS 532 does not send a consistent NAS IP address and, in these circumstances, another attribute (e.g., Called Station Id or Provider Id) is used to associate the access session with the service provider or ISP. Such cases reduce the usefulness of the NAS IP for duplicate detection.

[000176] Duplicate accounting records may be sent by the "batch" providers whose accounting feeds are assumed to be duplicate-free. Duplicate accounting records may be manually injected into the settlement system 474 when batches of records are sent by real-time accounting providers to complete their accounting responsibility when they have failed to deliver accounting records for one of the reasons described above. In these cases, arbitrary datasets may be sent by the service providers 452, which must be specially processed by personnel at the access broker system 454 to prepare them for submission in a data normalization process. Such datasets may contain data describing both previously reported sessions as well as the missing sessions for which the correction is attempted. Because these record batches are typically preprocessed as one-offs, little control exists to prevent duplicate injection. It will be appreciated that this process can be automated in view of the unique session identification.

[000177] Some service providers may admit duplicate transaction data records to the access broker system 454 due to irregular use of key duplicate fields. For example, in certain circumstances, service providers fill the NAS IP attribute with random data that thus adversely influences the duplicate filter criteria.

Other anomalies such as inconsistent session id generation or a failure to fix session duration at the time of user disconnect may generate duplicates that appear to correspond to distinct sessions. Once again, the unique session identification can assist in resolving these problems.

[000178] As shown by way of example in this document, duplicate accounting/transaction records are actively filtered by the settlement system 474 using an algorithm that matches six "key" fields of each real-time accounting/ transaction data record against all other real-time accounting/transaction data records that are received within the previous 30 days. Using the unique session_id field that uniquely identifies each approved single session, enhanced accuracy may be obtained.

Duplicate Alias Records

[000179] Duplicate alias records arise when an algorithm to detect duplicates inappropriately identifies a record as duplicate. For example, such cases can arise when a service provider's NAS (for example the NAS 532 of the ISP 510 in Figure 6) does not generate or reuses session identification data values within a short time. In addition to, or instead of, any session identification generated by the service provider that is not reliable, the unique session identification of the embodiment of the present invention, which uniquely defines each single access session, may be used by the duplicate detection algorithm to reduce the occurrence of duplicate alias records. In particular, the modified unique session identification in the session_id tables may be used to at least substantially reduce duplicate alias records as each session is uniquely identified.

Invalid Session-Length Records

[000180] It will be appreciated that all accounting/transaction data records received by the settlement system 474 relating to the duration of an access session may not always be complete. For example, an accounting/transaction data record may have session time duration data (e.g. an Acct-Session-Time attribute) missing, contain a zero value, contain an inaccurate value for the session (e.g., reporting a session as being 4 minutes long when it was in fact 3 minutes long), contain an unreasonably large value or is invalid as defined by RFC 2139, section 5.7 and so on. Invalid access time duration may occur, for

example, when a modem bank of a service provider does not report disconnection by the user and the NAS 532 continues to accumulate session time until another session starts on the same physical modem port or a timeout occurs for some other reason.

[000181] Accounting/transaction data records with a duplicate invalid session-length can arise for a variety of reasons, for example:

Missing Acct-Session-Time

[000182] When an accounting/transaction data record is received by the netserver 470 and is missing the Acct-Session-Time attribute, the netserver 470 typically sends on an accounting/transaction data record with a zero session length.

Inaccurate Acct-Session-Time

[000183] Inaccurate time accounting by the NAS 532 or intentional fraud by a service provider can generate accounting/transaction data records with inaccurate session durations.

Acct-Session-Time of Zero

[000184] When an accounting/transaction data record with a zero value Session-Time attribute is received by the netserver 470, the netserver 470 typically sends on an accounting/transaction data record with a zero session length.

Large Acct-Session-Time

[000185] Due to fraud, malfunction or inappropriate configuration, session time accounting may identify sessions of extravagant duration.

Disconnect Detection Failure

[000186] "Long" sessions or multiple sessions with identical duration are sometimes due to malfunction and/or inappropriate configuration of the modem bank of the service provider, which fails to detect user, disconnect for extended periods.

Fraudulent Access

[000187] Extended session times may also be due to continuous use by malicious users.

Corrupted Acct-Session-Time

[000188] Errors in the field handling by the NAS 532, the authorization/authentication server 600 of the service provider, or the netserver 470 of the service provider may corrupt the session time attribute of an accounting/transaction data record. Based on actual samples, this occurs often when long, vendor-specific data are present in some preceding RADIUS packet.

Genuine Long Sessions

[000189] The accuracy of the filtering of long sessions is dependent upon the filter threshold (typically about 100 hours).

[000190] Enhanced accuracy may be achieved by correlating, associating or the like the session length provided in the SQM records with session length provided in the accounting records. As each transaction data record has its unique session identification, the session length may be obtained from an associated record using the session_id field of the acct_trans tables and session_id field of the SQM tables 860. Data missing in one transaction data record can thus be obtained from another transaction data record bearing the unique session identification.

Overlapping accounting records

[000191] In certain circumstances, transaction data records are received from service providers that include the same user credentials (e.g. the same user name and password) that overlap in time. In the present embodiment of the invention, as each access session includes a unique session identification, analysis of overlapping transaction data records may be facilitated. In particular, the session_id field of the acct_trans table 854 and the session_id field of the SQM table 860 may be used for determining the session details of these records. For example, using the unique session identification, it may be determined if such sessions are two genuine different sessions or if the sessions are generated due to a faulty NAS 532.

Disputed Records

[000192] As the session identification uniquely identifies each single access session, and data related to the session including the unique session identification is sent to various different servers in the system, dispute resolution may be facilitated. In particular, a customer support team could compare the session details in the authentication or authorized transactions, accounting and SQM tables 856, 854 and 860 respectfully thereby providing three different sources of transaction data uniquely associated with a single user access session. The session_id field of the tables facilitates association, correlation or the like to corroborate details of the particular user access session.

Challenge Provider Records Recording Quality

[000193] As each session is uniquely identified, and various different servers communicate transaction data records independently to the settlement system 474, the quality of transaction data records received from a particular service provider may be evaluated by comparing transaction data records from that particular service provider with records from other sources. This may assist a network access team in isolating problems that relate to the accounting function as such or relate to technology problems at the service provider.

Legitimate Id Usage By More Than One Person

[000194] As the unique session identification uniquely identifies each single user session, it may be used to identify separate user access occurrences in which legitimate use of a single user identification data is used by more that one user. Thus, 456 customers may share the same login name, e.g. because the organization is small and/or chooses to operate in such a fashion. In such instances, it is possible to see logins from multiple locations with coincidental start times and session lengths being the same. The inclusion of the unique session identification is used to investigate these situations with enhanced accuracy.

Policy Management Versioning Of Dialer

[000195] The unique session identification can be used to associate, correlate, or the like SQM records with accounting records whereby accounting records

created without SQM records can be used to reveal the use of connection application (e.g., the connection application 466) technology, or non-supported versions thereof, that are not associated or provided by the access broker system 454. Typically, a report is created of customers and individual users who are using non-supported versions of the connection application 466 (e.g., connect dialer technology) thereby to help to migrate such users to a more current version. In certain circumstances, when an inappropriate connection application 466 is user by a customer 456, an account associated with the customer may be automatically disabled. Accordingly, the customer 456 may then be forced to contact the access brokerage system 454 to identify the problem and thereby force a version migration.

Overall Billing Process Quality Improvement

[000196] It will thus be appreciated that the inclusion of the unique session identification, that uniquely identifies all transaction data records associated with a single user session, enhances the accuracy of transaction record processing. Accordingly, less billing disputes are likely to arise and any dispute resolution that may arise may be settled more expeditiously.

[000197] Figure 13 is a diagram of a computer system 700, which may be configured as a network access device, such as 205, 305 or 405, a network dialer 466, or a netserver, such as 240, 350, 468, or 472. Computer system 700 includes a processor 750 operatively connected to random access memory (RAM) 735 and read only memory (ROM) 740 via a system bus 745. A processor 750 is also connected to a fixed disk 720, which may be an optical disk or other storage medium, through an input/output bus 730. Alternatively, the processor 750 may be connected to multiple storage devices through the input/output bus 730. The processor 750 communicates data using the system bus 745 and the input/output bus 730.

[000198] The system bus 745 and the input/output bus 730 may also receive inputs from a keypad 725 or an input device 710. The system bus 745 and the input/output bus 730 may provide outputs to a display 705, the fixed disk 720, and/or the output device 715. The memory and storage media 735, 740 may also include a flash memory, EEPROM, or any combination of the above.

[000199] The computer system 700 may be controlled by operating system software, which includes a file management system, such as, a disk operating system, which is part of the operating system software. The file management system may be stored in non-volatile storage device, such as the ROM 740, and may be configured to cause the processor 750 to execute the various functions required by the operating system to input and output data and to store data in the RAM 735 and on the ROM 740. For one embodiment of exemplary computer system 700, instructions may be stored on the fixed disk 720 or in the ROM 740 that cause the processor 750 to perform the functions of a network access device, such as the network access device 205 or 305. In an alternative embodiment, instructions may be stored on the fixed disk 720 or the ROM 740 that cause the processor 750 to perform the functions of a network decryption server, such as the netservers 240, 350, 472 or 468.

[000200] Thus, a method and system for associating a plurality of transaction data records generated in a service access system is described. In the foregoing detailed description, the invention has been described with reference to specific exemplary embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader scope and spirit of the invention as set forth in the appended claims. The specification and drawings are, accordingly, to be regarded in an illustrative sense rather than a restrictive sense.

CLAIMS

What is claimed is:

1. A method of associating a plurality of transaction data records generated in a service access system including at least one service provider, the transaction data records being generated in response to a user accessing the system during a single user session and method including:

generating a unique session identification that is uniquely associated with the single user session and which is receivable by the at least one service provider, the unique session identification being included in the transaction data record;

receiving the plurality of transaction data records at a transaction processing facility from the at least one service provider; and

processing the transaction data records using the unique session identification of each transaction data record.

2. The method of claim 1, which includes providing the unique session identification in a user identification string of each transaction data record when the user session is authorized.

3. The method of claim 2, which includes generating a unique code that is uniquely associated with the single user session and including the unique code in the user identification string.

4. The method of claim 3, which includes generating the unique code at a connection application via which the user requests access and combining the unique code with a connection application identification which identifies the connection application.

5. The method of claim 4, which includes generating the unique code by means of a counter.

6. The method of claim 3, which includes providing the unique session identification in a format suitable for communication using a protocol from one of Point-to-Point Protocol (PPP), Password Authentication Protocol (PAP), Challenge-Handshake Authentication Protocol (CHAP), Remote Authentication Dial In User Service (RADIUS) protocol, Terminal Access Controller Access Control System (TACACS) protocol, Lightweight Directory Access Protocol (LDAP), NT Domain authentication protocol, Unix password authentication protocol, HyperText Transfer Protocol (HTTP), HyperText Transfer Protocol over Secure sockets layer (HTTPS), Extended Authentication Protocol (EAP), Transport Layer Security (TLS) protocol, Token Ring protocol and Secure Remote Password protocol (SRP).

7. The method of claim 6, which includes providing the unique session identification within a user string with a maximum length of 63 characters.

8. The method of claim 6, which includes randomly generating three alphanumeric digits to define the unique code; providing a five-digit connection application identification that uniquely identifies the connection application; and providing an eleven-character user identification string that identifies the user.

9. The method of claim 6, which includes constructing a modified transaction record data from at least one of the unique session identification, a customer identification from the authenticating service provider, customer data, customer routing data for internal customer routing, user identification data, customer domain data used by the user for internal routing, and non routing data customer data of a transaction data record.

10. The method of claim 3, which includes allowing commencement of the session only if the user is positively authenticated by the system, and upon positive authentication, storing the unique session identification in a

session identification field in an authentication transaction storage area at a transaction processing facility.

11. The method of claim 10, which includes retrieving transaction data records from the at least one service provider and storing each transaction data record in an accounting transaction storage area based on the unique session identification data.

12. The method of claim 11, which includes periodically receiving batch loading transaction data records from the at least one service provider; constructing modified session identification data records; and storing the modified session identification data records in a session identification field in a batch history storage area.

13. The method of claim 12, which includes receiving Service Quality Monitor (SQM) transaction data records; constructing modified transaction data records from the SQM data records; and storing the modified transaction data records in a session identification field in an SQM storage area.

14. The method of claim 11, which includes comparing session identification data in the authentication transaction storage area and the accounting transaction area to identify missing accounting records.

15. The method of claim 11, which includes identifying each transaction data record without a unique session identification in the accounting transaction storage area to identify transaction data records provided by the at least one service provider which were not authenticated.

16. The method of claim 11, which includes searching for duplicate session identification data in the accounting transaction storage area to identify duplicate transaction data records.

17. The method of claim 11, which includes using the unique session identification to identify at least one of duplicate alias records, ISDN dual-channel records, invalid session length records, and overlapping accounting records.

18. A system for processing transaction data records generated in a service access system including at least one service provider, the transaction data records being generated in response to a user accessing the system during a single user session and system including:

a session identification generator to generate a unique session identification that is uniquely associated with the single user session and which is receivable by the at least one service provider, the unique session identification being included in the transaction data record; and

a transaction processing facility to process the plurality of transaction data records received from the at least one service provider using the unique session identification of each record.

19. The system of claim 18, in which the session identification generator provides the unique session identification data in a user identification string of each transaction data record when the user session is authorized.

20. The system of claim 19, in which the session identification generator generates a unique code that is uniquely associated with the single user session, the unique code being included in the user identification string.

21. The system of claim 20, in which, the session identification generator is implemented by a software application on a connection application via which the user requests access and the unique code is combined with a connection application identification which identifies the connection application.

22. The system of claim 21, in which the session identification generator is a counter that generates the unique code.

23. The system of claim 20, which includes a protocol from one of Point-to-Point Protocol (PPP), Password Authentication Protocol (PAP), Challenge-Handshake Authentication Protocol (CHAP), Remote Authentication Dial In User Service (RADIUS) protocol, Terminal Access Controller Access Control System (TACACS) protocol, Lightweight Directory Access Protocol (LDAP), NT Domain authentication protocol, Unix password authentication protocol, HyperText Transfer Protocol (HTTP), HyperText Transfer Protocol over Secure sockets layer (HTTPS), Extended Authentication Protocol (EAP), Transport Layer Security (TLS) protocol, Token Ring protocol and Secure Remote Password protocol (SRP), the unique session identification being provided in a format suitable for communication using.

24. The system of claim 23, which provides the unique session identification within a user string with a maximum length of 63 characters.

25. The system of claim 24, in which the unique code is provided by three alphanumeric digits randomly generated; the connection application identification is provided by five-digits that uniquely identifies the connection application; and the user identification string that identifies the user is provided by eleven-characters.

26. The system of claim 20, which allows commencement of the session only if the user is positively authenticated the by system, and upon positive authentication, the transaction processing facility stores the unique session identification in a session identification field in an authentication transaction storage area.

27. The system of claim 26, in which the transaction processing facility receives transaction data records from the at least one service provider

and stores each transaction data record in an accounting transaction storage area based on the unique session identification data.

28. The system of claim 27, in which the transaction processing facility periodically receives batch loading transaction data records from the at least one service provider; constructs modified session identification data records; and stores the modified session identification data records in a session identification field in a batch history storage area.

29. The system of claim 28, in which the transaction processing facility receives Service Quality Monitor (SQM) transaction data records; constructs modified transaction data records from the SQM transaction data records; and stores the modified transaction data records in a session identification field in an SQM storage area.

30. A method of processing a plurality of transaction data records generated in a service access system including at least one service provider, method including:

receiving the transaction data records from the at least one service provider, each transaction data record being generated in response to a user accessing the system during a single user session; and

identifying transaction data records associated with the single user session based on a session identification included in each transaction data record, each session identification uniquely identifying a single user session.

31. The method of claim 30, which includes identifying the unique session identification in a user identification string of each transaction record.

32. The method of claim 31, which includes identifying a unique code that is uniquely associated with the single user session and included in the user identification string.

33. The method of claim 32, including identifying a connection application identification that identifies a connection application via which the user requests access and that has generated the unique code.

34. The method of claim 32, which receives the transaction data record including the unique session identification using a protocol from one of Point-to-Point Protocol (PPP), Password Authentication Protocol (PAP), Challenge-Handshake Authentication Protocol (CHAP), Remote Authentication Dial In User Service (RADIUS) protocol, Terminal Access Controller Access Control System (TACACS) protocol, Lightweight Directory Access Protocol (LDAP), NT Domain authentication protocol, Unix password authentication protocol, HyperText Transfer Protocol (HTTP), HyperText Transfer Protocol over Secure sockets layer (HTTPS), Extended Authentication Protocol (EAP), Transport Layer Security (TLS) protocol, Token Ring protocol and Secure Remote Password protocol (SRP).

35. The method of claim 34, which extracts the unique session identification from a user string with a maximum length of 63 characters.

36. The method of claim 35, which identifies the unique code from three alphanumeric digits; identifies the connection application from a five-digit connection application identification; and identifies the user from an eleven-character user identification string.

37. The method of claim 34, which includes constructing a modified transaction record data from at least one of the unique session identification, a customer identification from the authenticating service provider, customer data, customer routing data for internal customer routing, user identification data, customer domain data used by the user for internal routing, and non routing data customer data of a transaction data record.

38. The method of claim 32, which includes storing the unique session identification in a session identification field in an authentication transaction storage area at a transaction processing facility.

39. The method of claim 32, which includes storing each transaction data record received from the service providers in an accounting transaction storage area based on the unique session identification data.

40. The method of claim 39, which includes periodically receiving batch loading transaction data records from the at least one service provider; constructing modified session identification data records from the transaction data records; and storing the modified session identification data records in a session identification field in a batch history storage area.

41. The method of claim 39, which includes receiving Service Quality Monitor (SQM) transaction data records; constructing modified transaction data records from the modified transaction data records; and storing the modified transaction data records in a session identification field in an SQM storage area.

42. The method of claim 39, which includes comparing session identification data in the authentication transaction storage area and the accounting transaction area to identify missing accounting records.

43. The method of claim 39, which includes identifying each transaction data record without a unique session identification in the accounting transaction storage area to identify transaction data records provided by the at least one service provider which were not authenticated.

44. The method of claim 39, which includes searching for duplicate session identification data in the accounting transaction storage area to identify duplicate transaction records.

45. The method of claim 39, which includes using the unique session identification to identify at least one of duplicate alias records, ISDN dual-channel records, invalid session length records, and overlapping accounting records.

46. A transaction processing facility for processing a plurality of transaction data records generated in a service access system including at least one service provider, transaction processing facility arranged to:

receive the transaction data records from the at least one service provider, each transaction data record being generated in response to a user accessing the system during a single user session; and

identify transaction data records associated with the single user session based on a session identification included in each transaction data record, each session identification uniquely identifying a single user session.

47. The transaction processing facility of claim 46, which identifies the unique session identification in a user identification string of each transaction record.

48. The transaction processing facility of claim 47, which identifies a unique code that is uniquely associated with the single user session and included in the user identification string.

49. The transaction processing facility of claim 47, in which the transaction processing facility identifies a connection application identification that identifies a connection application that generated the unique code and via which the user gains access.

50. The transaction processing facility of claim 48, which receives the transaction data record including the unique session identification using a protocol from one of Point-to-Point Protocol (PPP), Password Authentication Protocol (PAP), Challenge-Handshake Authentication Protocol (CHAP),

Remote Authentication Dial In User Service (RADIUS) protocol, Terminal Access Controller Access Control System (TACACS) protocol, Lightweight Directory Access Protocol (LDAP), NT Domain authentication protocol, Unix password authentication protocol, HyperText Transfer Protocol (HTTP), HyperText Transfer Protocol over Secure sockets layer (HTTPS), Extended Authentication Protocol (EAP), Transport Layer Security (TLS) protocol, Token Ring protocol and Secure Remote Password protocol (SRP).

51. The transaction processing facility of claim 50, which extracts the unique session identification from a user string with a maximum length of 63 characters.

52. The transaction processing facility of claim 51, which identifies the unique code from three alphanumeric digits; identifies the connection application from a five-digit connection application identification; and identifies the user from an eleven-character user identification string.

53. The transaction processing facility of claim 47, which constructs a modified transaction record data from at least one of the unique session identification, a customer identification from the authenticating service provider, customer data, customer routing data for internal customer routing, user identification data, customer domain data used by the user for internal routing, and non routing data customer data of a transaction data record.

54. The transaction processing facility of claim 47, which stores the unique session identification in a session identification field in an authentication transaction storage area at the transaction processing facility.

55. The transaction processing facility of claim 47, which stores each transaction data record received from the at least one service provider in an accounting transaction storage area based on the unique session identification data.

56. The transaction processing facility of claim 55, which periodically receives batch loading transaction data records from the at least one service provider from which it constructs modified session identification data records that are stored in a session identification field in a batch history storage area.

57. The transaction processing facility of claim 55, which receives Service Quality Monitor (SQM) transaction data records and constructs modified transaction data records from the SQM transaction data records and stores them in a session identification field in an SQM storage area.

58. The transaction processing facility of claim 55, which compares session identification data in the authentication transaction storage area and the accounting transaction area to identify missing accounting records.

59. The transaction processing facility of claim 55, which identifies each transaction data record without a unique session identification in the accounting transaction storage area to identify transaction data records provided by the at least one service provider which were not authenticated.

60. The transaction processing facility of claim 55, which searches for duplicate session identification data in the accounting transaction storage area to identify duplicate transaction records.

61. The transaction processing facility of claim 55, which uses the unique session identification to identify at least one of duplicate alias records, ISDN dual-channel records, invalid session length records, and overlapping accounting records.

62. A method of connecting a user to an access service provider, the method including creating a unique session identification associated with a single user session during which the user accesses the service provider, the

unique session identification being provided in a user identification string of each transaction data record when the user session is authorized.

63. The method of claim 62, which includes generating a unique code that is uniquely associated with the single user session and including the unique code in the user identification string.

64. The method of claim 63, including generating the unique code at a connection application via which the user gains access and combining the unique code with a connection application identification which identifies the connection application.

65. The method of claim 64, which includes generating the unique code by means of a counter.

66. The method of claim 63, which includes providing the unique session identification in a format suitable for communication using a protocol from one of Point-to-Point Protocol (PPP), Password Authentication Protocol (PAP), Challenge-Handshake Authentication Protocol (CHAP), Remote Authentication Dial In User Service (RADIUS) protocol, Terminal Access Controller Access Control System (TACACS) protocol, Lightweight Directory Access Protocol (LDAP), NT Domain authentication protocol, Unix password authentication protocol, HyperText Transfer Protocol (HTTP), HyperText Transfer Protocol over Secure sockets layer (HTTPS), Extended Authentication Protocol (EAP), Transport Layer Security (TLS) protocol, Token Ring protocol and Secure Remote Password protocol (SRP).

67. The method of claim 66, which includes providing the unique session identification within a user string with a maximum length of 63 characters.

68. The method of claim 67, which includes randomly generating three alphanumeric digits to define the unique code; providing a five-digit connection application identification that uniquely identifies the connection application; and providing an eleven-character user identification string that identifies the user.

69. Connection apparatus for connecting a user to an access service provider, the apparatus including a session identification generator that creates a unique session identification associated with each session that is authorized.

70. The connection apparatus of claim 69, in which the session identification generator generates a unique code that is uniquely associated with the single user session, the unique code being included in the user identification string.

71. The connection apparatus of claim 70, in which the session identification generator combines the unique code with a connection apparatus identification that identifies the connection application.

72. The connection apparatus of claim 70, in which the session identification generator generates the unique code by means of a counter.

73. The connection apparatus of claim 70, in which the unique session identification is provided in a format suitable for communication using a protocol from one of Point-to-Point Protocol (PPP), Password Authentication Protocol (PAP), Challenge-Handshake Authentication Protocol (CHAP), Remote Authentication Dial In User Service (RADIUS) protocol, Terminal Access Controller Access Control System (TACACS) protocol, Lightweight Directory Access Protocol (LDAP), NT Domain authentication protocol, Unix password authentication protocol, HyperText Transfer Protocol (HTTP), HyperText Transfer Protocol over Secure sockets layer (HTTPS), Extended

Authentication Protocol (EAP), Transport Layer Security (TLS) protocol, Token Ring protocol and Secure Remote Password protocol (SRP).

74. The connection apparatus of claim 73, in which the unique session identification is provided within a user string with a maximum length of 63 characters.

75. The connection apparatus of claim 74, in which the session identification generator randomly generates three alphanumeric digits to define the unique code; provides a five-digit connection application identification that uniquely identifies the connection apparatus; and provides for an eleven-character user identification string to identify the user.

76. A machine-readable medium embodying a sequence of instructions that associate a plurality of transaction data records generated in a service access system including at least one service provider, the transaction data records being generated in response to a user accessing the system during a single user session, the instructions, when executed by a machine, cause the machine to:

generate a unique session identification that is uniquely associated with the single user session and which is receivable by the at least one service provider, the unique session identification being included in the transaction data record;

receive the plurality of transaction data records at a transaction processing facility from the service providers; and

process the transaction data records using the unique session identification of each transaction data record.

77. The machine-readable medium of claim 76, in which the unique session identification is provided in a user identification string of each transaction data record when the user session is authorized.

78. The machine-readable medium of claim 77, in which a unique code that is uniquely associated with the single user session is generated and included in the user identification string.

79. A machine-readable medium embodying a sequence of instructions that processes a plurality of transaction data records generated in a service access system including at least one service provider, the instructions, when executed by a machine, cause the machine to:

receive the transaction data records from the at least one provider, each transaction data record being generated in response to a user accessing the system during a single user session; and

identify transaction data records associated with the single user session based on a session identification included in each transaction data record, each session identification uniquely identifying a single user session.

80. A machine-readable medium embodying a sequence of instructions that relate to connecting a user to an access provider, the instructions, when executed by a machine, cause the machine to create a unique session identification associated with a single user session during which the user accesses the service provider, the unique session identification being provided in a user identification string of each transaction data record when the user session is authorized.

81. A transaction processing facility for processing a plurality of transaction data records generated in a service access system including at least one service provider, transaction processing facility including:

receiver means for receiving the transaction data records from the at least one service provider, each transaction data record being generated in response to a user accessing the system during a single user session; and

processor means to identify transaction data records associated with the single user session based on a session identification included in each transaction data record, each session identification uniquely identifying a single user session.

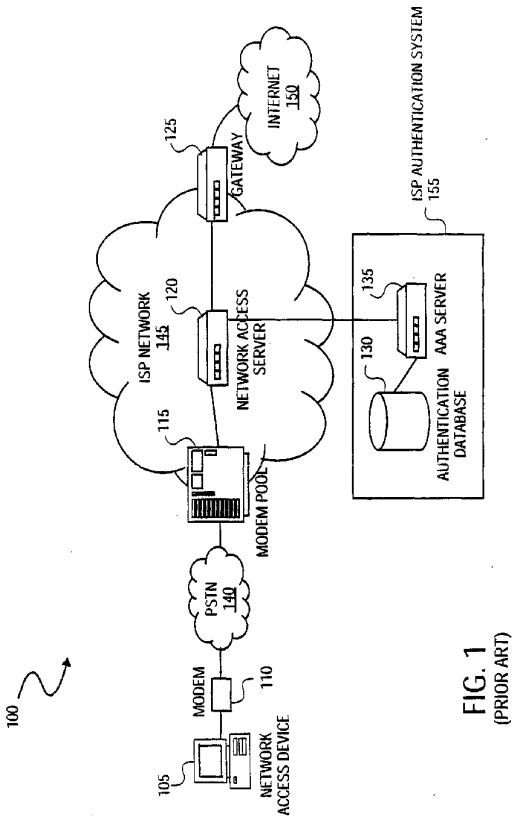


FIG. 1
(PRIOR ART)

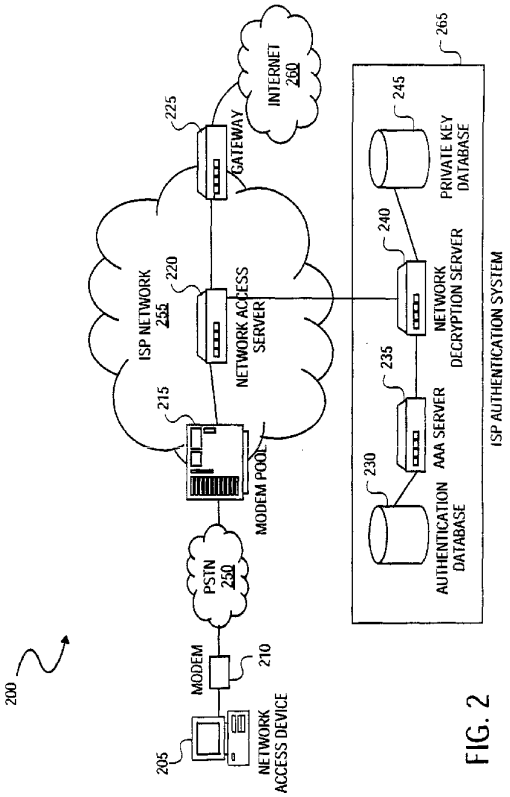

200
205
NETWORK ACCESS DEVICE
210
MODEM
PSTN 250
215
MODEM POOL
ISP NETWORK 255
220
NETWORK ACCESS SERVER
225
GATEWAY
INTERNET 260
265
ISP AUTHENTICATION SYSTEM
230
AUTHENTICATION DATABASE
235
AAA SERVER
240
NETWORK DECRYPTION SERVER
245
PRIVATE KEY DATABASE

FIG. 2

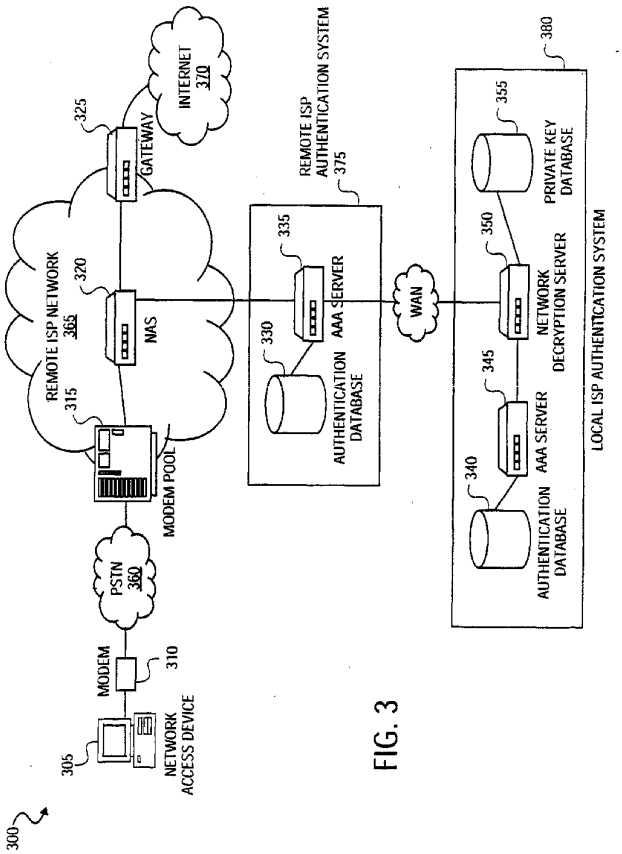

300
NETWORK ACCESS DEVICE
305
MODEM
310
PSTN
360
MODEM POOL
REMOTE ISP NETWORK
365
315
NAS
320
GATEWAY
325
INTERNET
370
REMOTE ISP AUTHENTICATION SYSTEM
375
AAA SERVER
335
AUTHENTICATION DATABASE
330
WAN
LOCAL ISP AUTHENTICATION SYSTEM
380
NETWORK DECRYPTION SERVER
350
PRIVATE KEY DATABASE
355
AAA SERVER
345
AUTHENTICATION DATABASE
340

FIG. 3

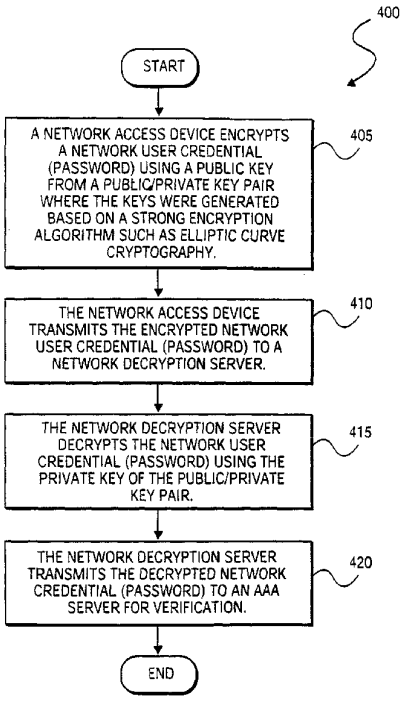

400
START
A NETWORK ACCESS DEVICE ENCRYPTS A NETWORK USER CREDENTIAL (PASSWORD) USING A PUBLIC KEY FROM A PUBLIC/PRIVATE KEY PAIR WHERE THE KEYS WERE GENERATED BASED ON A STRONG ENCRYPTION ALGORITHM SUCH AS ELLIPTIC CURVE CRYPTOGRAPHY.
405
THE NETWORK ACCESS DEVICE TRANSMITS THE ENCRYPTED NETWORK USER CREDENTIAL (PASSWORD) TO A NETWORK DECRYPTION SERVER.
410
THE NETWORK DECRYPTION SERVER DECRYPTS THE NETWORK USER CREDENTIAL (PASSWORD) USING THE PRIVATE KEY OF THE PUBLIC/PRIVATE KEY PAIR.
415
THE NETWORK DECRYPTION SERVER TRANSMITS THE DECRYPTED NETWORK CREDENTIAL (PASSWORD) TO AN AAA SERVER FOR VERIFICATION.
420
END

FIG. 4

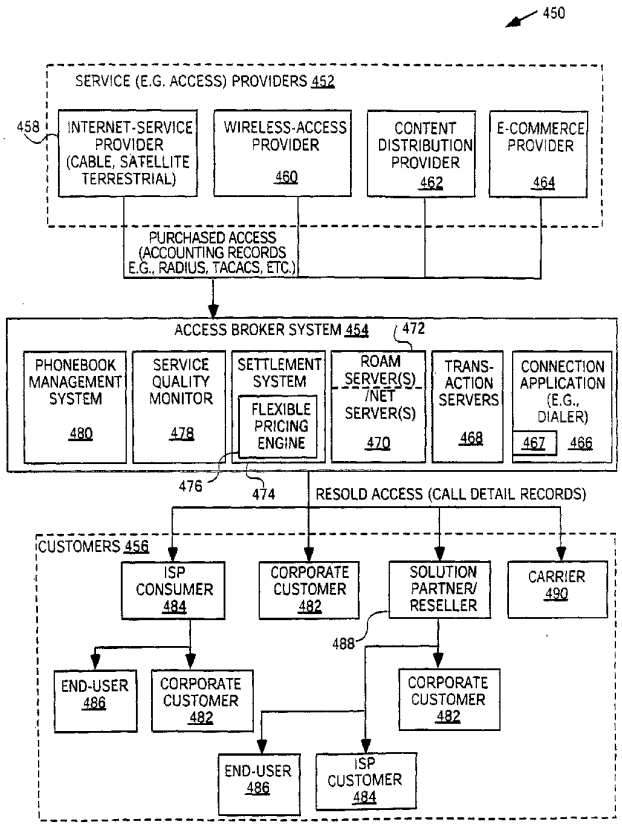

450
SERVICE (E.G. ACCESS) PROVIDERS 452
458
INTERNET-SERVICE PROVIDER (CABLE, SATELLITE TERRESTRIAL)
WIRELESS-ACCESS PROVIDER 460
CONTENT DISTRIBUTION PROVIDER 462
E-COMMERCE PROVIDER 464
PURCHASED ACCESS (ACCOUNTING RECORDS E.G., RADIUS, TACACS, ETC.)
ACCESS BROKER SYSTEM 454
472
PHONEBOOK MANAGEMENT SYSTEM 480
SERVICE QUALITY MONITOR 478
SETTLEMENT SYSTEM
FLEXIBLE PRICING ENGINE
ROAM SERVER(S) /NET SERVER(S) 470
TRANS-ACTION SERVERS 468
CONNECTION APPLICATION (E.G., DIALER) 467 466
476
474
RESOLD ACCESS (CALL DETAIL RECORDS)
CUSTOMERS 456
ISP CONSUMER 484
CORPORATE CUSTOMER 482
SOLUTION PARTNER/ RESELLER
CARRIER 490
488
END-USER 486
CORPORATE CUSTOMER 482
CORPORATE CUSTOMER 482
END-USER 486
ISP CUSTOMER 484

FIG. 5

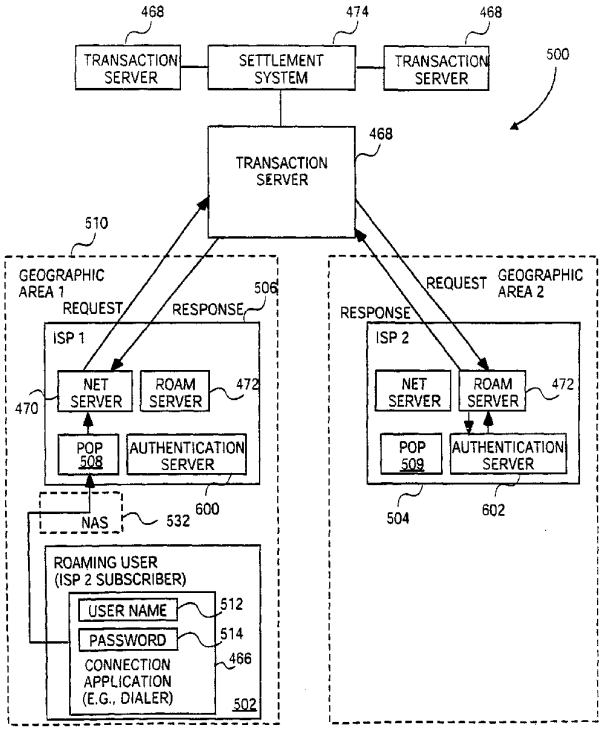

468
474
468
TRANSACTION SERVER
SETTLEMENT SYSTEM
TRANSACTION SERVER
500
468
TRANSACTION SERVER
510
GEOGRAPHIC AREA 1
REQUEST
RESPONSE
506
ISP 1
NET SERVER
ROAM SERVER
472
470
POP 508
AUTHENTICATION SERVER
600
NAS
532
ROAMING USER (ISP 2 SUBSCRIBER)
USER NAME
512
PASSWORD
514
CONNECTION APPLICATION (E.G., DIALER)
466
502
REQUEST
GEOGRAPHIC AREA 2
RESPONSE
ISP 2
NET SERVER
ROAM SERVER
472
POP 509
AUTHENTICATION SERVER
504
602

FIG. 6

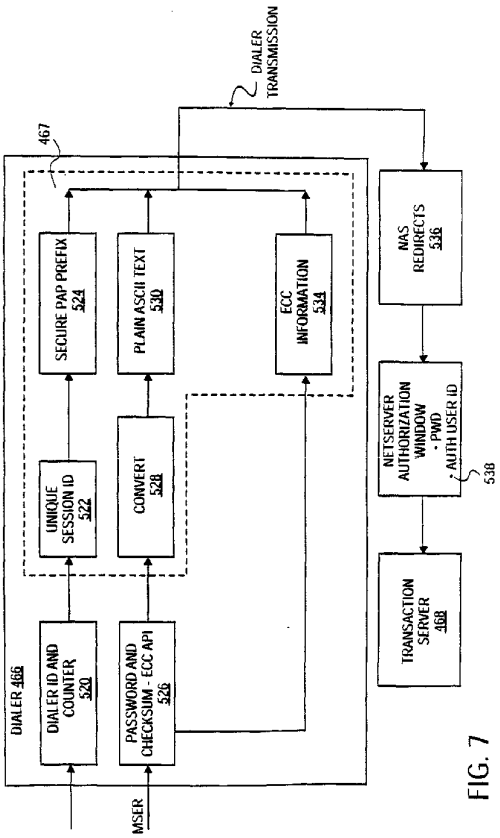

DIALER TRANSMISSION
467
SECURE PAP PREFIX 524
PLAIN ASCII TEXT 530
ECC INFORMATION 534
NAS REDIRECTS 536
UNIQUE SESSION ID 522
CONVERT 528
NETSERVER AUTHORIZATION WINDOW · PWD · AUTH USER ID
538
DIALER 466
DIALER ID AND COUNTER 520
PASSWORD AND CHECKSUM - ECC API 526
TRANSACTION SERVER 468
MSER

FIG. 7

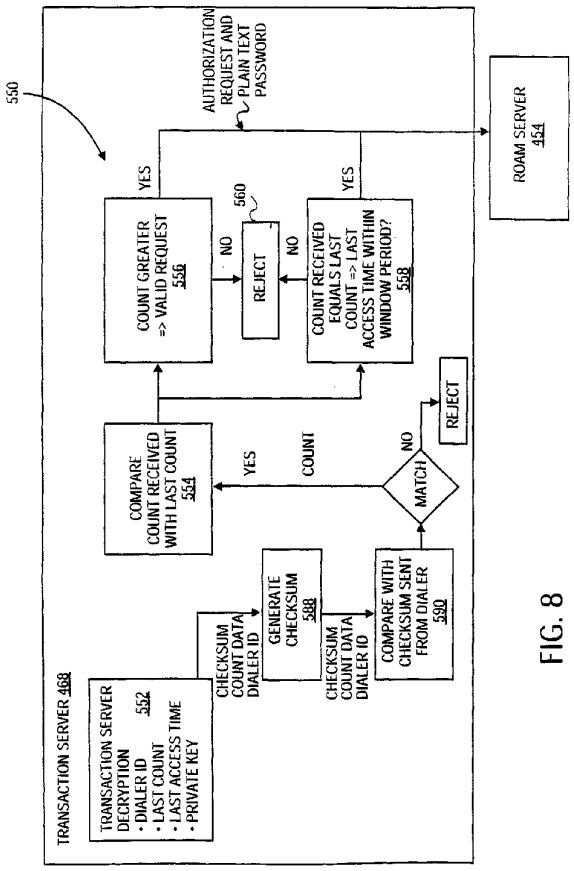

550
AUTHORIZATION REQUEST AND PLAIN TEXT PASSWORD
ROAM SERVER 454
YES
COUNT GREATER => VALID REQUEST 556
NO
560
REJECT
NO
COUNT RECEIVED EQUALS LAST COUNT => LAST ACCESS TIME WITHIN WINDOW PERIOD? 558
YES
COMPARE COUNT RECEIVED WITH LAST COUNT 564
YES
COUNT
MATCH
NO
REJECT
TRANSACTION SERVER 468
TRANSACTION SERVER DECRYPTION
• DIALER ID 562
• LAST COUNT
• LAST ACCESS TIME
• PRIVATE KEY
CHECKSUM COUNT DATA DIALER ID
GENERATE CHECKSUM 588
CHECKSUM COUNT DATA DIALER ID
COMPARE WITH CHECKSUM SENT FROM DIALER 590

FIG. 8

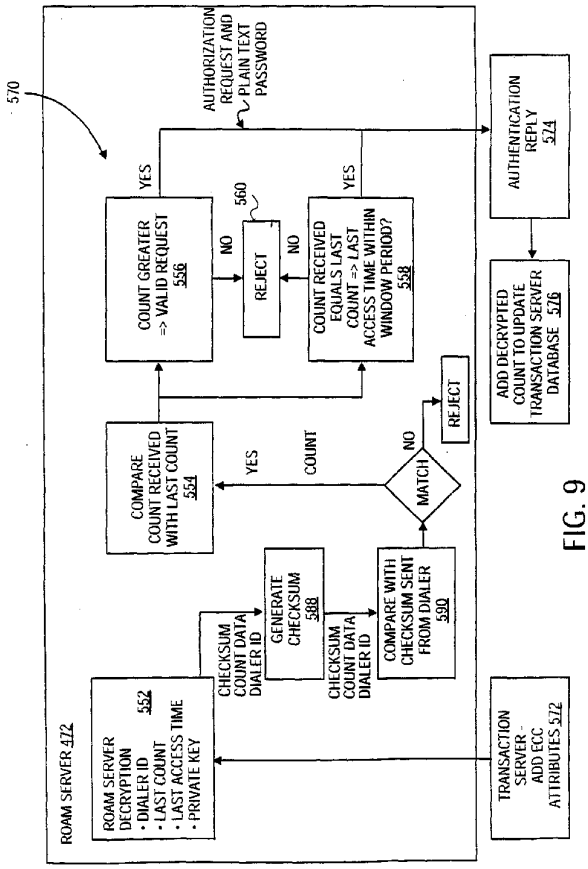

570
ROAM SERVER 472
ROAM SERVER DECRYPTION 552
• DIALER ID
• LAST COUNT
• LAST ACCESS TIME
• PRIVATE KEY
CHECKSUM COUNT DATA DIALER ID
GENERATE CHECKSUM 588
CHECKSUM COUNT DATA DIALER ID
COMPARE WITH CHECKSUM SENT FROM DIALER 590
MATCH
NO
REJECT
YES
COUNT
COMPARE COUNT RECEIVED WITH LAST COUNT 554
COUNT GREATER => VALID REQUEST 556
NO
560
REJECT
NO
COUNT RECEIVED EQUALS LAST COUNT => LAST ACCESS TIME WITHIN WINDOW PERIOD? 558
YES
YES
AUTHORIZATION REQUEST AND PLAIN TEXT PASSWORD
TRANSACTION SERVER ADD ECC ATTRIBUTES 572
ADD DECRYPTED COUNT TO UPDATE TRANSACTION SERVER DATABASE 576
AUTHENTICATION REPLY 574

FIG. 9

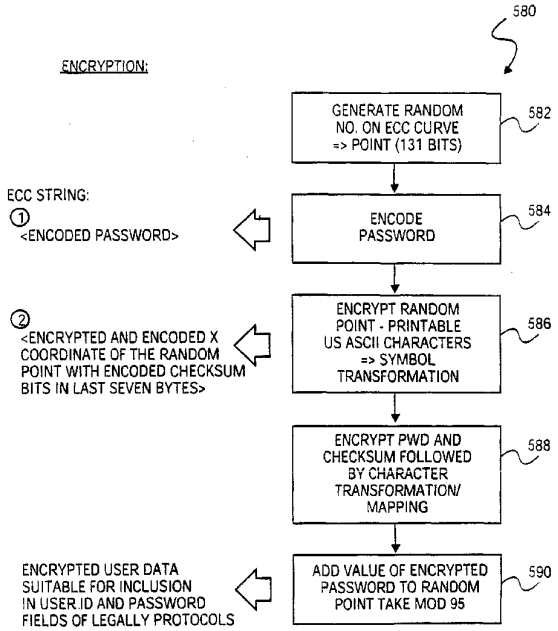

580
ENCRYPTION:
GENERATE RANDOM NO. ON ECC CURVE => POINT (131 BITS)
582
ECC STRING:
① <ENCODED PASSWORD>
ENCODE PASSWORD
584
② <ENCRYPTED AND ENCODED X COORDINATE OF THE RANDOM POINT WITH ENCODED CHECKSUM BITS IN LAST SEVEN BYTES>
ENCRYPT RANDOM POINT - PRINTABLE US ASCII CHARACTERS => SYMBOL TRANSFORMATION
586
ENCRYPT PWD AND CHECKSUM FOLLOWED BY CHARACTER TRANSFORMATION/ MAPPING
588
ENCRYPTED USER DATA SUITABLE FOR INCLUSION IN USER ID AND PASSWORD FIELDS OF LEGALLY PROTOCOLS
ADD VALUE OF ENCRYPTED PASSWORD TO RANDOM POINT TAKE MOD 95
590

FIG. 10

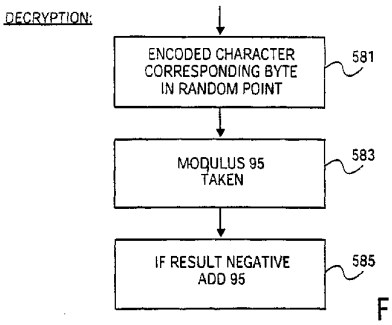

DECRYPTION:
ENCODED CHARACTER CORRESPONDING BYTE IN RANDOM POINT
581
MODULUS 95 TAKEN
583
IF RESULT NEGATIVE ADD 95
585

FIG. 11

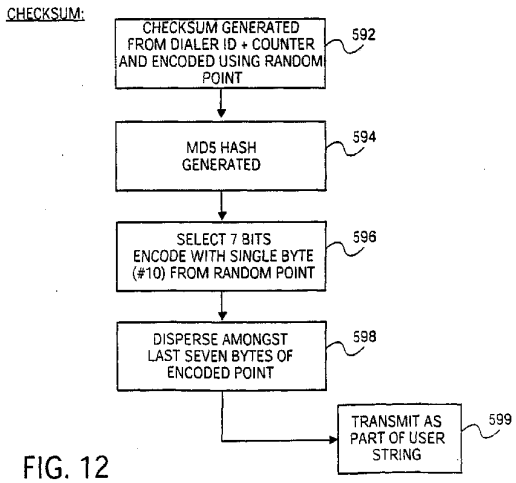

CHECKSUM:
CHECKSUM GENERATED FROM DIALER ID + COUNTER AND ENCODED USING RANDOM POINT
592
MD5 HASH GENERATED
594
SELECT 7 BITS ENCODE WITH SINGLE BYTE (#10) FROM RANDOM POINT
596
DISPERSE AMONGST LAST SEVEN BYTES OF ENCODED POINT
598
TRANSMIT AS PART OF USER STRING
599

FIG. 12

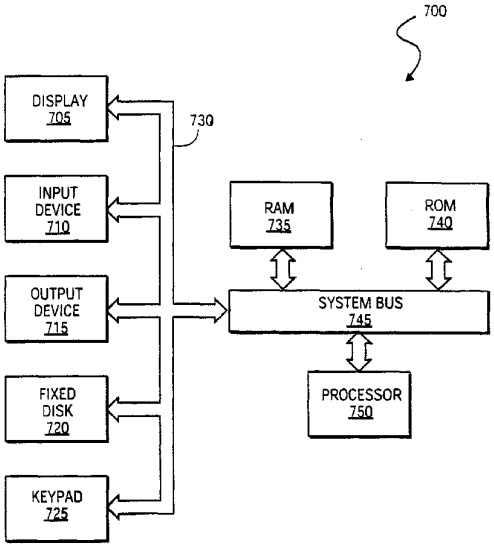

700
DISPLAY
705
730
INPUT
DEVICE
710
RAM
735
ROM
740
OUTPUT
DEVICE
715
SYSTEM BUS
745
FIXED
DISK
720
PROCESSOR
750
KEYPAD
725

FIG. 13

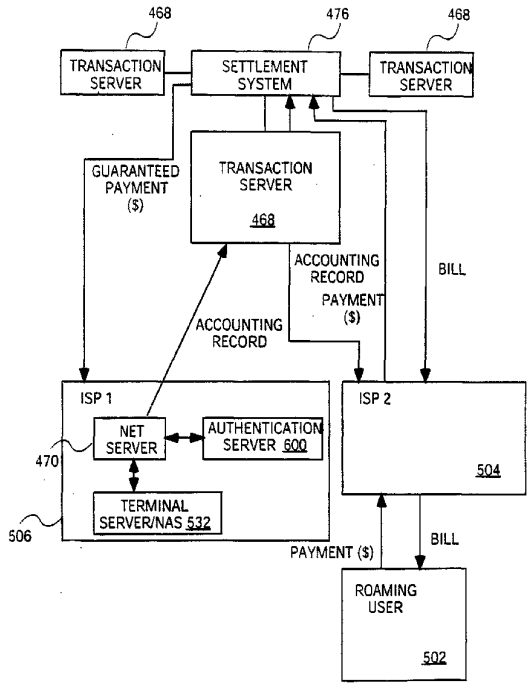

468
476
468
TRANSACTION SERVER
SETTLEMENT SYSTEM
TRANSACTION SERVER
GUARANTEED PAYMENT ($)
TRANSACTION SERVER
468
ACCOUNTING RECORD
PAYMENT ($)
BILL
ACCOUNTING RECORD
ISP 1
ISP 2
NET SERVER
AUTHENTICATION SERVER 600
470
504
TERMINAL SERVER/NAS 532
506
PAYMENT ($)
BILL
ROAMING USER
502

FIG. 14

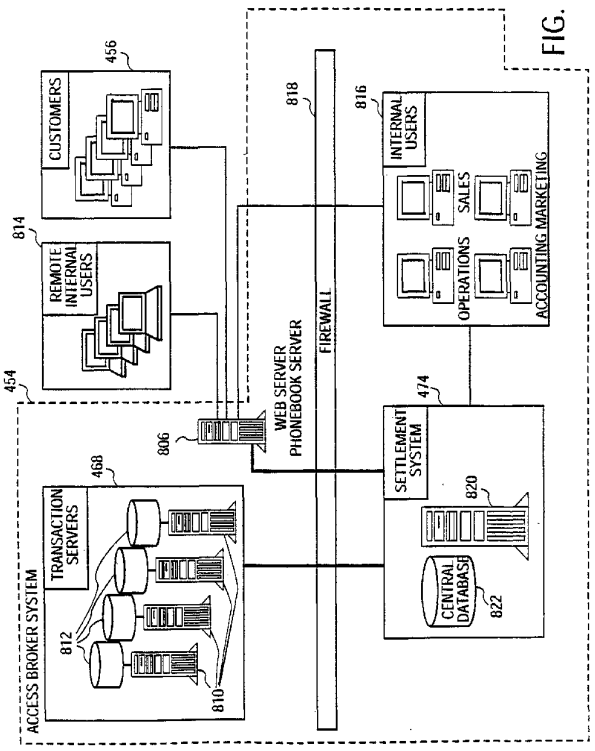

456
CUSTOMERS
814
REMOTE INTERNAL USERS
454
ACCESS BROKER SYSTEM
468
TRANSACTION SERVERS
812
810
806
WEB SERVER PHONEBOOK SERVER
818
FIREWALL
816
INTERNAL USERS
SALES
OPERATIONS
ACCOUNTING MARKETING
474
SETTLEMENT SYSTEM
820
CENTRAL DATABASE
822

FIG. 15

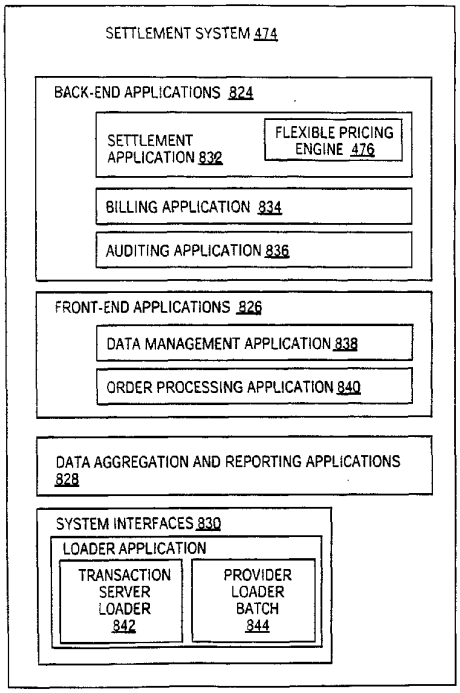

SETTLEMENT SYSTEM 474
BACK-END APPLICATIONS 824
SETTLEMENT APPLICATION 832
FLEXIBLE PRICING ENGINE 476
BILLING APPLICATION 834
AUDITING APPLICATION 836
FRONT-END APPLICATIONS 826
DATA MANAGEMENT APPLICATION 838
ORDER PROCESSING APPLICATION 840
DATA AGGREGATION AND REPORTING APPLICATIONS 828
SYSTEM INTERFACES 830
LOADER APPLICATION
TRANSACTION SERVER LOADER 842
PROVIDER LOADER BATCH 844

FIG. 16

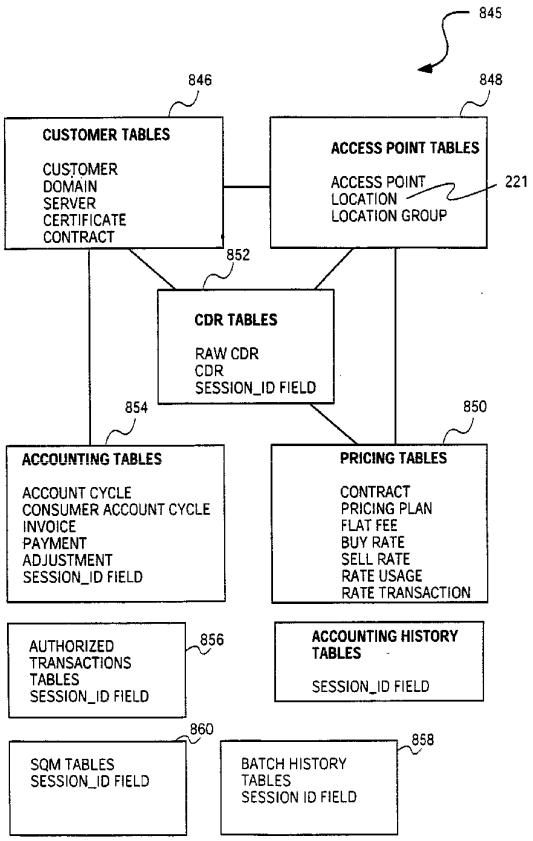
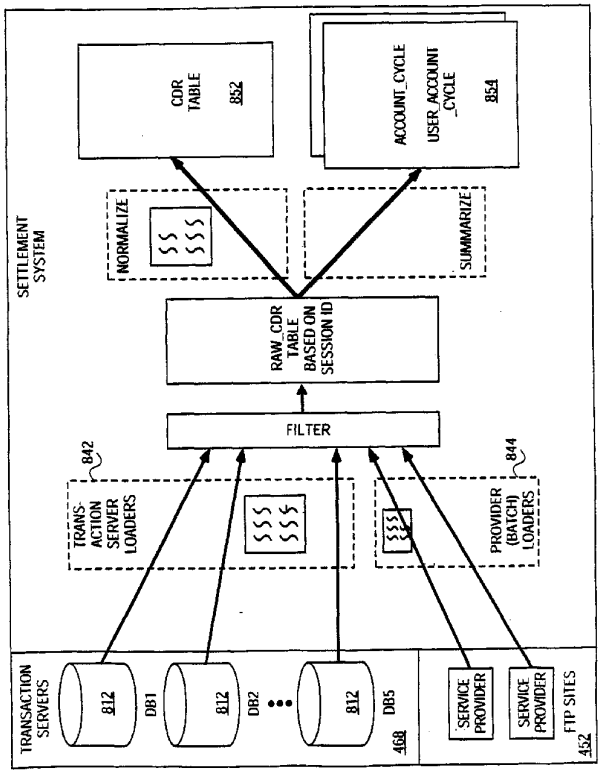

845
846
848
CUSTOMER TABLES
CUSTOMER
DOMAIN
SERVER
CERTIFICATE
CONTRACT
ACCESS POINT TABLES
ACCESS POINT
LOCATION
LOCATION GROUP
221
852
CDR TABLES
RAW CDR
CDR
SESSION_ID FIELD
854
850
ACCOUNTING TABLES
ACCOUNT CYCLE
CONSUMER ACCOUNT CYCLE
INVOICE
PAYMENT
ADJUSTMENT
SESSION_ID FIELD
PRICING TABLES
CONTRACT
PRICING PLAN
FLAT FEE
BUY RATE
SELL RATE
RATE USAGE
RATE TRANSACTION
AUTHORIZED
TRANSACTIONS
TABLES
SESSION_ID FIELD
856
ACCOUNTING HISTORY
TABLES
SESSION_ID FIELD
860
858
SQM TABLES
SESSION_ID FIELD
BATCH HISTORY
TABLES
SESSION ID FIELD

FIG. 17A

FIG. 17B


SETTLEMENT SYSTEM
CDR TABLE 852
ACCOUNT_CYCLE
USER_ACCOUNT _CYCLE 854
NORMALIZE
SUMMARIZE
RAW_CDR TABLE BASED ON SESSION ID
FILTER
842
TRANS-ACTION SERVER LOADERS
844
PROVIDER (BATCH) LOADERS
TRANSACTION SERVERS
812
DB1
812
DB2
812
DB5
468
SERVICE PROVIDER
SERVICE PROVIDER
FTP SITES
452

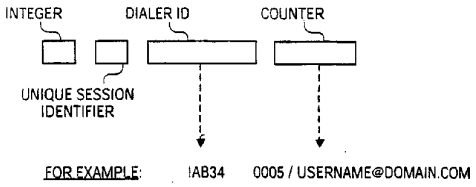

INTEGER
DIALER ID
COUNTER
UNIQUE SESSION IDENTIFIER
FOR EXAMPLE:
!AB34
0005 / USERNAME@DOMAIN.COM

FIG. 18

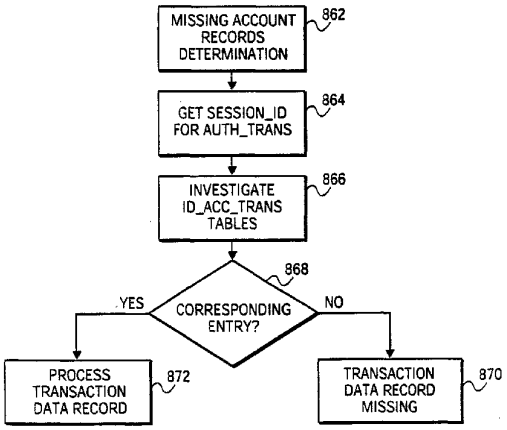

MISSING ACCOUNT RECORDS DETERMINATION
862
GET SESSION_ID FOR AUTH_TRANS
864
INVESTIGATE ID_ACC_TRANS TABLES
866
868
CORRESPONDING ENTRY?
YES
NO
PROCESS TRANSACTION DATA RECORD
872
TRANSACTION DATA RECORD MISSING
870

FIG. 19

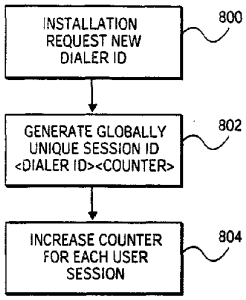

INSTALLATION REQUEST NEW DIALER ID
800
GENERATE GLOBALLY UNIQUE SESSION ID <DIALER ID><COUNTER>
802
INCREASE COUNTER FOR EACH USER SESSION
804

FIG. 20

【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US02/12475

A. CLASSIFICATION OF SUBJECT MATTER
IPC( ) : G06F 01/24
US CL : 713/153, 169, 182, 200, 201
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
U.S. : 713/153, 169, 182, 200, 201

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
West

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 6,023,470 (LEE et al.) 8 February 2000, col. 6, lines 34-55, col. 14, lines 24-43, col. 53, lines 55-63.	1-81

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
30 July 2002 (30.07.2002)	11 SEP 2002
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703)305-3230	Authorized officer Thomas R. Peeso Telephone No. 703 305-3900

Form PCT/ISA/210 (second sheet) (July 1998)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US02/12475

Continuation of Item 4 of the first sheet:
Title is too long.
METHOD AND SYSTEM FOR ASSOCIATING DATA RECORDS

Form PCT/ISA/210 (second sheet) (July 1998)

フロントページの続き

(81)指定国 AP(GH,GM,KE,LS,MW,MZ,SD,SL,SZ,TZ,UG,ZM,ZW),EA(AM,AZ,BY,KG,KZ,MD,RU,TJ,TM),EP(AT,BE,CH,CY,DE,DK,ES,FI,FR,GB,GR,IE,IT,LU,MC,NL,PT,SE,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN,GQ,GW,ML,MR,NE,SN,TD,TG),AE,AG,AL,AM,AT,AU,AZ,BA,BB,BG,BR,BY,BZ,CA,CH,CN,CO,CR,CU,CZ,DE,DK,DM,DZ,EC,EE,ES,FI,GB,GD,GE,GH,GM,HR,HU,ID,IL,IN,IS,JP,KE,KG,KP,KR,KZ,LC,LK,LR,LS,LT,LU,LV,MA,MD,MG,MK,MN,MW,MX,MZ,NO,NZ,OM,PH,PL,PT,RO,RU,SD,SE,SG,SI,SK,SL,TJ,TM,TN,TR,TT,TZ,UA,UG,US,UZ,VN,YU,ZA,ZM,ZW

Ｆターム(参考) 5B085 AC04 AE02 AE03 BC01
5K030 KA01 MC08