República Federativa do Brasil
Ministério da Economia
Instituto Nacional da Propriedade Industrial

(11) BR 112015027175-8 B1

(22) Data do Depósito: 30/04/2014

(45) Data de Concessão: 11/01/2022

(54) Título: MÉTODO PARA SINCRONIZAR UM CONJUNTO DE CREDENCIAIS DE SENHA ENTRE UM SERVIÇO DE ORIGEM E UM SERVIÇO ALVO, E DISPOSITIVO DE ARMAZENAMENTO LEGÍVEL POR COMPUTADOR

(51) Int.Cl.: G06F 21/31; H04L 9/32.

(30) Prioridade Unionista: 30/04/2013 US 13/873,882.

(73) Titular(es): MICROSOFT TECHNOLOGY LICENSING, LLC.

(72) Inventor(es): ARIEL N. GORDON; JONATHAN M. LUK; RAMAN N. CHIKKAMAGALUR; ZIAD ELMALKI; SERGII GUBENKO; GIRISH CHANDER; ANANDHI SOMASEKARAN; MURLI D. SATAGOPAN.

(86) Pedido PCT: PCT US2014036004 de 30/04/2014

(87) Publicação PCT: WO 2014/179386 de 06/11/2014

(85) Data do Início da Fase Nacional: 27/10/2015

(57) Resumo: MÉTODO PARA SINCRONIZAR UM CONJUNTO DE CREDENCIAIS DE SENHA ENTRE UM SERVIÇO DE ORIGEM E UM SERVIÇO ALVO E DISPOSITIVO DE ARMAZENAMENTO LEGÍVEL POR COMPUTADOR. A presente invenção refere-se a seguramente sincronizar senhas que são mudadas em uma localização de origem (por exemplo, um serviço de diretório de escritório) para uma localização alvo (por exemplo, um serviço de diretório de nuvem), de modo que as mesmas credenciais possam ser utilizadas para logar na localização de origem e alvo, porém sem necessariamente tendo cada controlador de domínio tratando a sincronização. A senha de texto puro não é revelada, ao invés, utilizando valores de hash computados desta para representar os dados relativos à senha. O alvo pode receber um hash secundário de um hash primário, e por meio disto somente receber armazenar um blob de senha. A autenticação é executada utilizando os mesmos algoritmos de hash no serviço alvo para computar um blob e comparar em relação ao blob sincronizado. Também descrita está a agilidade de criptografia e/ou mudar os algoritmos de hash sem requerer uma mudança de senha de usuário.

Relatório Descritivo da Patente de Invenção para **"MÉTODO PARA SINCRONIZAR UM CONJUNTO DE CREDENCIAIS DE SENHA ENTRE UM SERVIÇO DE ORIGEM E UM SERVIÇO ALVO, E DISPOSITIVO DE ARMAZENAMENTO LEGÍVEL POR COMPUTADOR".**

ANTECEDENTES

[0001] Cada vez mais organizações estão utilizando aplicações e recursos de serviço de nuvem em oposição a somente utilizar aplicações e recursos local, (onde "local" refere-se a sob o controle da organização, independentemente de qualquer localização física, em contraste com a nuvem). Como com as aplicações e recursos local, os usuários precisam de credenciais para acessar serviços de nuvem existentes. Note que algumas (tipicamente muito pequenas) organizações somente utilizam a nuvem para a sua infraestrutura e aplicações de identidade baseada em credencial, e assim utilizam a nuvem para lidar com a autenticação baseada em credencial.

[0002] Organizações muito grandes administram um serviço de diretório local (um exemplo do qual é Active Directory® da Microsoft Corporation que inclui seus servidores de controlador de domínio) para autenticar usuários, e para aplicações para descobrir contas de usuário e relações entre contas. Entre outras coisas, isto permite que tais organizações retenham controle total de seus dados relativos a credenciais para propósitos de segurança, ao invés de prover os dados para a nuvem. Grandes organizações utilizam o que (por exemplo, em um cenário de Active Directory®) pode ser referido como um serviço de federação/federação, o qual contém mecanismos para usuários individuais alavancarem as suas credenciais local para acessar recursos na nuvem. As credenciais não são sincronizadas; ao invés, a nuvem direciona solicitações de login e similares para uma infraestrutura de identidade local para autenticação, permitindo um usuário somente

entrar uma vez.

[0003] No entanto, uma federação é relativamente muito dispendiosa para instalar e manter, e assim somente grandes organizações tendem a utilizar uma federação. Muitas pequenas organizações desejam utilizar o mesmo nome de usuário e senha para acessar os recursos e aplicações local assim como recursos e aplicações de nuvem. Sem a federação, no entanto, algum modo para lidar com as credenciais local e as credenciais de nuvem é necessário.

[0004] Uma solução é interceptar a senha de usuário de texto puro para transporte para um serviço de diretório alvo. A senha de usuário de texto puro pode ser replicada para todos os servidores/banco de dados na infraestrutura de identidade. No entanto, isto pode ser inseguro, especificamente quando o serviço de diretório de nuvem é um alvo. Mais ainda, software precisa ser configurado em cada servidor no serviço de diretório alvo para capturar todos os eventos de mudança de senha de usuário. Entre outras desvantagens, isto é ineficiente e inconveniente para manter.

[0005] Muitas companhias não querem liberar os dados de credencial local para a nuvem por razões de segurança, o que cria um problema de autenticação. Uma solução é emitir um conjunto de credenciais para os usuários acessarem as aplicações de nuvem, e outro conjunto de credenciais para os usuários acessarem as aplicações local. Isto é também ineficiente e inconveniente para manter.

SUMÁRIO

[0006] Este Sumário está provido para introduzir uma seleção de conceitos representativos em uma forma simplificada que são adicionalmente abaixo descritos na Descrição Detalhada. Este Sumário não pretende identificar as características chave ou características essenciais do assunto reivindicado, nem pretende ser utilizado em nenhum modo que limitaria o escopo do assunto reivindicado.

[0007] Resumidamente, vários aspectos do assunto aqui descrito estão direcionados para seguramente sincronizar senhas que são mudadas em uma localização de origem para uma localização alvo, de modo que as mesmas credenciais possam ser utilizadas na localização de origem e alvo. Em um aspecto, um valor de hash que é computado com base em uma senha de texto puro é recebido, no qual o valor de hash foi computado em resposta a um evento de mudança de senha em um serviço de origem. Os dados que correspondem ao valor de hash são exportados para um serviço alvo, para sincronizar a nova senha para o serviço alvo para utilização em autenticação de identidade. Os dados que correspondem ao valor de hash podem ser secundariamente executados hash em um blob protegido por senha utilizando um algoritmo de hash secundário.

[0008] Em um aspecto, um processo de hospedeiro de sincronização está acoplado a uma malha de domínio. O processo de hospedeiro de sincronização está configurado para sincronizar as mudanças de senha recebidas na malha de domínio com um serviço de diretório alvo externo à malha, (por exemplo, um serviço de diretório de nuvem). O processo de hospedeiro de sincronização obtém um valor de hash representativo de uma senha de texto puro da malha de domínio, processa o valor de hash em um blob protegido por segredo através pelo menos um algoritmo de hash secundário, exporta o blob protegido por segredo para o serviço de diretório alvo. O processo de hospedeiro de sincronização pode ser acoplado e obter o valor de hash de um componente da malha ou acoplado na malha, na qual o componente está configurado para receber os dados de mudança de senha replicados que correspondem a uma mudança de senha feita em qualquer controlador de domínio da malha.

[0009] Em um aspecto, uma pluralidade de conjuntos de dados compreendem blobs protegidos que correspondem a senhas de texto

puro é mantida. Cada blob está associado com uma identidade, na qual os blobs são computados de uma senha de texto puro por pelo menos dois algoritmos de hash. Outro blob que é computado com outro algoritmo de hash está associado com uma identidade, incluindo substituindo o blob por outro blob. Isto pode ser realizado computando o blob com outro algoritmo de hash para cada identidade, incluindo para cada identidade, executar o hash no blob associado com esta identidade para o outro blob para esta identidade. Isto também pode ser realizado recebendo o outro blob de um componente de serviço de diretório local, e recebendo informações que identificam as informações que correspondem ao outro algoritmo de hash.

[00010] Outras vantagens podem tornar-se aparentes da descrição detalhada seguinte quando tomada em conjunto com os desenhos.

BREVE DESCRIÇÃO DOS DESENHOS

[00011] A presente invenção está ilustrada como exemplo e não limitada nas figuras acompanhantes nas quais números de referência iguais indicam elementos similares e nas quais:

[00012] Figura 1 é um diagrama de blocos que representa componentes exemplares configurados para sincronizar mudança de senhas feitas em um serviço de diretório local para um serviço de diretório de nuvem, de acordo com uma implementação exemplar.

[00013] Figura 2 é um fluxograma que representa etapas exemplares que podem ser tomadas para seguramente sincronizar uma mudança de senha em um serviço de diretório de origem para um serviço de diretório alvo, de acordo com uma implementação exemplar.

[00014] Figura 3 é um diagrama de blocos que representa componentes exemplares configurados para sincronizar as mudanças de senha de serviço de diretório local feitas em qualquer controlador de domínio para um serviço de diretório de nuvem através de um único componente, de acordo com uma implementação exemplar.

[00015] Figura 4 é um fluxograma que representa etapas exemplares que podem ser tomadas através de toda uma operação de sincronização de senha segura, de acordo com uma implementação exemplar.

[00016] Figura 5 é um fluxograma que representa etapas exemplares que podem ser tomadas para autenticar um usuário durante uma tentativa de logon utilizando dados relativos a senha seguramente sincronizados, de acordo com uma implementação exemplar.

[00017] Figura 6 é um fluxograma que representa etapas exemplares que podem ser tomadas para mudar um algoritmo de hash secundário e dados relativos a senha mantidos para um conjunto de usuários, de acordo com uma implementação exemplar.

[00018] Figura 7 é um diagrama de blocos que representa ambientes de rede não limitantes exemplares nos quais várias modalidades aqui descritas podem ser implementadas.

[00019] Figura 8 é um diagrama de blocos que representa um sistema de computação não limitante exemplar ou ambiente de operação no qual um ou mais aspectos de várias modalidades aqui descritas podem ser implementados.

DESCRIÇÃO DETALHADA

[00020] Vários aspectos da tecnologia aqui descrita estão geralmente direcionados para uma tecnologia de sincronização de senha que permite que um único conjunto de credenciais seja utilizado tanto para acesso de recursos local e acesso de recurso de nuvem. Como será compreendido, a tecnologia provê uma solução que é relativamente direta de instalar e manter no escritório, enquanto ao mesmo tempo sendo segura.

[00021] Em um aspecto, um agente de sincronização desempenha as operações de sincronização de um serviço de diretório local com um serviço de diretório de nuvem. Em uma implementação, o agente

de sincronização pode ser adicionado a uma malha de controlador de domínio como um único componente (por exemplo, executando em uma única máquina unida a um domínio) em oposição a executar em cada controlador de domínio na malha de domínio.

[00022] Em um aspecto, as credenciais mantidas no serviço de diretório local são sincronizadas com o serviço de diretório de nuvem primeiro utilizando um ou mais algoritmos de hash para executar um hash nas senhas. Um hash primário é utilizado, e pode ser utilizado em combinação com pelo menos um hash secundário. As senhas de texto puro nunca são enviadas para a nuvem.

[00023] Em um aspecto, a tecnologia suporta tendo o sistema local mudando para um novo algoritmo de hash primário, sem requerer que os usuários mudem as suas senhas existentes ou de outro modo recapturem as senhas de texto puro dos usuários. Ainda, se o algoritmo de hash secundário for comprometido ou um algoritmo de hash secundário mais seguro de outro modo torna-se desejável de utilizar, o algoritmo de hash secundário pode ser mudado sem requerer que os usuários mudem as suas senhas existentes ou de outro modo recapturem as senhas de texto puro dos usuários.

[00024] Deve ser compreendido que qualquer dos exemplos aqui não é limitante. Por exemplo, muitos dos exemplos aqui são geralmente descritos em um ambiente de serviço de diretório tal como Active Directory®; no entanto qualquer infraestrutura/ambiente de identidade similar pode se beneficiar na tecnologia aqui descrita. Mais ainda, apesar dos exemplos serem direcionados para uma sincronização de credencial segura, outros de dados que precisam ser seguramente sincronizadas podem se beneficiar da tecnologia aqui descrita. Como tal, a presente invenção não está limita a quaisquer modalidade, aspectos, conceitos, estruturas, funcionalidades ou exemplos específicos aqui descritos. Ao invés, qualquer uma das modalidades, aspectos,

conceitos, estruturas, funcionalidades ou exemplos aqui descritos são não limitantes, e a presente invenção pode ser utilizada em vários modos que proveem benefícios e vantagens em sincronização de dados, segurança de dados e/ou serviços de nuvem em geral.

[00025] A Figura 1 é um diagrama de blocos que mostra componentes exemplares que podem ser utilizados para seguramente sincronizar dados incluindo dados de credencial de componentes local para um serviço de diretório de nuvem 102. Os componentes local 104 incluem um processo de hospedeiro de sincronização 106 (por exemplo, uma máquina de sincronização de gerenciador de identidade) que inclui um agente de sincronização de senha 108. Em geral, o processo de hospedeiro de sincronização 106 compreende um processo que ativamente aciona a recuperação e exportação de credenciais de um serviço de diretório de origem 110.

[00026] Em uma implementação, sincronização é realizada através do agente de sincronização de senha 108, o qual chama no serviço de diretório local (local) 110 (o diretório de origem) através de uma interface adequada 112 para obter dados relativos a credencial, os quais como abaixo descrito, compreendem senhas em hash. De modo a obter somente as senhas em hash mudadas (deltas) deste o último tempo de sincronização, a chamada pode prover uma estampa de tempo de sincronização. Por exemplo, Active Directory® publicamente documentou API (IDL_DRSGetNCChanges) que quando chamado recupera e retorna uma lista de mudanças desde que uma estampa de tempo provida, o qual é o último tempo de sincronização provido pelo agente de sincronização de senha 108. Em um cenário no qual os dados de mudança compreendem mais do que dados relativos a senha, o agente de sincronização 108 analisa/filtra os dados retornados para determinar o conjunto de credenciais atualizadas desde o último tempo de sincronização.

[00027] O conjunto de credenciais mudadas é retornado para o agente de sincronização de senha 108 como um conjunto de credenciais em hash. Em uma implementação, estes hashes não são persistidos pelo processo de hospedeiro de sincronização 106 ou o agente de sincronização de senha 108, e são somente temporariamente utilizados na tentativa de sincronizar o hash de credencial para um serviço de diretório alvo, por exemplo, o serviço de diretório de nuvem 102 na Figura 1. Em uma implementação, os hashes de senha local são secundariamente executado o hash utilizando um valor randomicamente gerado (salt) um número de iterações antes de serem enviados para o serviço de diretório alvo (nuvem) 102.

[00028] Em uma implementação, o agente de sincronização de senha 108 tenta sincronizar somente as credenciais de identidades em escopo (onde escopo é um conceito bem conhecido em infraestruturas de identidade) para o serviço de diretório alvo 102. As credenciais que pertencem a identidades fora de escopo não são sincronizadas para o serviço de diretório alvo 102. Ainda, as credenciais que pertencem a identidades que não foram provisionadas para o serviço de diretório alvo 102 também não são sincronizadas; ao invés estas podem ser sincronizadas em um tempo posterior quando esta identidade foi provisionada com sucesso no serviço de diretório alvo 102.

[00029] Na Figura 1, um componente de conector de diretório alvo representado como um agente de gerenciamento de nuvem 114 é responsável por lidar com a exportação da credencial em hash para a nuvem 116. Para este fim, um componente de interface inicial de nuvem 118 (por exemplo, servidor de interface inicial de sincronização de serviço de diretório) recebe a solicitação para atualizar a credencial, e então tenta persistir a credencial em hash no sistema de armazenamento de serviço de diretório alvo 120 através de uma interface programática (por exemplo, privada) 122. Se o hash de credencial for persistido com

sucesso no serviço de diretório alvo, um status de sucesso é retornado para o componente de interface inicial 118 e o componente de interface inicial 118 e retorna um status de sucesso para o processo de hospedeiro de sincronização 106. Quando do recebimento de uma resposta de "sucesso", o processo de hospedeiro de sincronização 106 considera a credencial sincronizada com sucesso para o serviço de diretório alvo 102. Se uma resposta de falha for encontrada, a exportação pode ser enfileirada para uma nova tentativa em um tempo posterior.

[00030] A Figura 2 mostra a operação acima como um conjunto de etapas exemplares. Algumas das etapas estão mostradas para uma única credencial, no entanto como pode prontamente ser apreciado, a sincronização de credencial pode ser em lote, e/ou algumas ou todas as etapas podem ser desempenhadas em paralelo.

[00031] Na etapa 202, o agente de sincronização de senha 108 (Figura 1) solicita e recebe as mudanças (desde uma dada estampa de tempo) do serviço de diretório de origem 110. A solicitação é feita em um tempo de sincronização, o qual pode ser periódico ou de outro modo. Como acima descrito, as senhas são colocadas em hash com uma função de hash primário, por exemplo, Ha(password), tal como MD4(password).

[00032] Quando recebendo as mudanças, como representado pela etapa 204, o agente de sincronização de senha 108 analisa as mudanças para determinar quais devem ser sincronizadas, por exemplo, são mudanças de senhas de identidades provisionadas, em escopo. Como acima mencionado, considere que somente uma credencial está sendo lidada neste tempo.

[00033] A etapa 206 representa colocar secundariamente em hash a senha em hash, por exemplo, H1(Ha(password)) tal como SHA256(MD4(password)). O hash secundário está adicionalmente abaixo descrito.

[00034] A etapa 208 exporta a credencial em hash para o serviço de diretório alvo 102, o qual tenta persisti-lo. A etapa 210 recebe o resultado da solicitação de exportação como um status retornado; se um sucesso for recebido como avaliado na etapa 212, a credencial foi sincronizada com sucesso para o serviço de diretório alvo (etapa 214) e o processo termina. Se uma falha for detectada através da etapa 212, a exportação é enfileirada para uma nova tentativa em um tempo posterior, como representado pela etapa 216.

[00035] Como geralmente representada na Figura 3, o a malha de domínio local 330 contém Domínio 1 (tendo domínio 1 controlador 1 - domínio 1 controlador j) até Domínio n (tendo domínio n controlador 1 - domínio n controlador k). Em um aspecto, a malha pode adicionar (por exemplo, pode ser unida por ou de outro modo acoplada a) um componente que executa em máquina ou similar que executa como serviço de localizador de controlador de domínio de serviço de diretório 332. Como é conhecido, as mudança de senhas são feitas em um controlador de domínio (por exemplo, o mais próximo do usuário, apesar de outros esquemas serem factíveis) e replicadas para outros controladores de domínio do domínio. Como aqui descrito, as senhas mudadas que foram executadas o hash com o hash primário são replicadas ao invés de senhas de texto puro.

[00036] O processo de hospedeiro de sincronização 106 contata o serviço de localizador de controlador de domínio de serviço de diretório 332 para determinar uma instância de controlador de domínio da qual recuperar os dados de mudança de credencial. Por exemplo, pode existir um controlador de domínio identificado em cada domínio para prover as mudanças no processo de hospedeiro de sincronização 106. Deste modo, o esquema de replicação existente de uma malha pode ser alavancado para desempenhar uma sincronização de mudança de senha com o serviço de nuvem; (note que isto está em con-

traste com os sistemas existentes nos quais componentes/extensão de código DLLs precisam ser registrados com todas as máquinas associadas com o diretório de origem/malha de modo a assegurar que todas as mudanças de credencial são capturadas e sincronizadas para o diretório alvo).

[00037] A Figura 4 resume as etapas exemplares relativas à implementação da Figura 3 e as operações de controlador de domínio em geral. A etapa 402 representa receber uma mudança de senha em texto puro, o qual está tipicamente no controlador de domínio mais próximo do usuário (apesar de outros esquemas, tal como baseado em balanceamento de carga, serem factíveis). A etapa 404 representa a senha sendo aplicada o hash neste controlador de domínio com o hash primário, por exemplo, Ha(Password). A etapa 406 representa replicar a senha em hash para os outros controladores de domínios.

[00038] A etapa 408 representa o processo de hospedeiro de sincronização 106 comunicando com o serviço de localizador de controlador de domínio de serviço de diretório 332 para determinar qual(is) controlador(es) de domínio contatar para os dados de mudança. Em geral, um controlador de domínio de cada domínio é identificado pelo serviço de localizador de controlador de domínio 332 para o processo de hospedeiro de sincronização 106.

[00039] A etapa 410 representa o agente de sincronização de senha recuperando os hashes de senha mudados de um controlador de domínio de serviço de diretório. Note que como uma alternativa, as mudanças podem ser empurradas para o agente de sincronização de senha para sincronização sob demanda ou alguma outra programação. É factível que o processo de hospedeiro de sincronização execute na mesma máquina que o serviço de localizador de controlador de domínio de serviço de diretório 332, apesar de que como acima descrito, o processo de hospedeiro de sincronização não persiste as se-

nhas em hash outro que conforme necessário para desempenhar a sincronização com o serviço alvo.

[00040] Apesar de ser factível para que o alvo seja sincronizado com e armazena a senha em hash, ter um hash secundário provê um número de benefícios como aqui descrito. A etapa 412 representa o hash secundário, por exemplo, Hl(Ha(password)). Em um aspecto, o hash secundário gera um blob de senha protegida o qual inclui o nome e versão de algoritmo de hash, mais salt randômico, contagem de iterações mais sinopse. The resultado deste hash secundário é sincronizado com (etapa 414) e armazenado (etapa 416) no serviço de diretório alvo. Note que a nuvem pode também desempenhar tal hash secundário, de modo a executar o hash mais uma vez antes do armazenamento.

[00041] Observando os aspectos de login, quando uma identidade tenta acessar um serviço ou software associado com o serviço de diretório alvo, por exemplo, através do servidor de interface inicial AuthN 124 (Figura 1) se a credencial for marcada como "sincronizada do diretório de origem" na plataforma de autenticação do serviço de diretório alvo, a plataforma de autenticação compreende executar o procedimento de verificação de login apropriado e comparar a credencial apresentada pela identidade contra o hash de credencial sincronizado do diretório de origem.

[00042] A plataforma de autenticação alvo está instruída para utilizar um algoritmo para corresponder ao algoritmo de hash local, mas este pode ser qualquer um algoritmo ou conjunto de algoritmos. Isto facilita um número de cenários, incluindo agilidade de criptografia. Em geral, a agilidade de criptografia permite que múltiplos algoritmos de hash sejam utilizados, e/ou combinações de algoritmos de hash. Como um resultado, o algoritmo de hash primário pode mudar ao longo do tempo, o algoritmo de hash secundário pode mudar ao longo do tem-

po, os algoritmos de diferentes serviços (por exemplo, de terceiros) podem ser utilizados, e assim por diante.

[00043] A Figura 5 mostra algumas etapas exemplares relativas a operações de logon no serviço de nuvem, iniciando na etapa 502 onde uma tentativa de logon com uma credencial é recebida. Se na etapa 504 a credencial não for marcada como "sincronizada do diretório de origem" ou similares, então por exemplo a nuvem está sendo acessada por um usuário que não faz parte de um serviço de diretório local, tal como um usuário de uma organização muito pequena que somente utiliza a nuvem para autenticação e acesso de recursos. Também, os usuários podem fazer parte do serviço de diretório local mas não utilizando a tecnologia aqui descrita, e assim a credencial não é marcada sincronizada. Se assim, a etapa 506 trata desta solicitação de outro modo, por exemplo, através de logon de nuvem convencional.

[00044] Se ao invés etapa 504 detectar que a credencial está marcada como "sincronizada do diretório de origem", a etapa 508 procura qual algoritmo de hash/dados utilizar, por exemplo, com base na identidade de usuário. A etapa 510 determina os parâmetros para este hash, por exemplo, salt e iterações. Note que em um cenário no qual somente um algoritmo de hash existe, as etapas 508 e 510 are não são necessárias, mas como pode ser prontamente apreciado, estas etapas proveem agilidade de criptografia.

[00045] A etapa 512 converte os dados de senha de logon para o blob de senha protegida, cuja etapa 514 compara contra o blob armazenado no banco de dados do serviço alvo. Se existir uma coincidência (etapa 516), o acesso é permitido através da etapa 518, de outro modo o acesso é negado através da etapa 520.

[00046] Note que a agilidade de criptografia suporta um novo algoritmo de hash (primário) local (Ha) sem impactar o serviço e sem precisar recapturar a senha de texto puro do usuário. Por exemplo, consi-

dere que o sistema local muda de Ha para Hb (por exemplo, a próxima versão do serviço de diretório desaprova MD4 em favor de alguma coisa mais moderna). Quaisquer novas senhas/senhas mudadas serão computados e sincronizadas como (H1(Hb(password))). No tempo de logon, quando os usuários digitam o seu nome de usuário e senha (texto puro), o sistema determina se (H1(Ha)) ou (Hl(Hb)) está presente no banco de dados, e aplica aquele apropriado na senha de texto puro para comparação.

[00047] Ainda, a plataforma de autenticação pode desempenhar um hash adicional de hashes armazenados conforme desejado. Isto facilita a proteção de dados resistentes ao tempo de senhas em repouso com agilidade de criptografia. Como exemplo, considere que o algoritmo de hash secundário (H1) is está comprometido, isto é, não mais considerado suficientemente seguro. O algoritmo de hash H1 pode ser efetiva amente substituído, sem precisar recapturar a senha de texto puro do usuário.

[00048] Como exemplo, considere que o blob de dados correntemente computado e armazenado é H1(Ha(password)). Para segurança, um novo algoritmo de hash secundário (H2) é introduzido. Como representado nas etapas 602, 604 e 606 da Figura 6, para cada usuário, o sistema alvo analisa o banco de dados inteiro, computa (H2(H1(Ha(password)))), e armazena o novo valor. Quando a análise está completa como avaliado pela etapa 608, o sistema apaga o (H1(Ha(password))) para todos os usuário na etapa 610 e muda para utilizar o algoritmo (H2(H1)). Assim, o sistema não mais armazena o hash em repouso compreendido. Note que é factível substituir o blob existente na etapa 606, no entanto se o processo de análise for extenso, os usuários podem ser impedidos de logar até que o processo de análise esteja completo.

[00049] No tempo de logon tudo funciona o mesmo como antes da

perspectiva do usuário. Conforme o usuário loga, o alvo determina que (H2(H1(Ha))) é o algoritmo de hash para computar o valor de hash para a senha provida e comparar o valor de hash com o que está armazenado.

[00050] O sistema alvo pode também mudar para outra função hash para novas senhas. Por exemplo, considere que outro algoritmo de hash H3 é desenvolvido que é considerado superior em algum modo àquele existente, por exemplo, H3 é muito melhor e/ou mais rápido do que H1. Neste exemplo a mudança não é um problema de segurança, e assim o (H1(Ha(password))) é seguro, e deixado intacto. O processo de hospedeiro de sincronização (e o serviço alvo) é atualizado para suportar (H3) para quaisquer novos usuários/senhas mudadas. Os usuários que mudam a sua senha são assim sincronizados utilizando (H3(Ha(password))). Os usuários que não mudam as suas senhas continuam a serem autenticados através do algoritmo (H1(Ha(password))).

[00051] O histórico de senha pode ser mantido no serviço de nuvem e utilizado no tempo de logon para evitar trancar usuários fora. Por exemplo, considere um usuário que mudou a sua senha em um dispositivo, resultando em sincronizar um blob para o serviço de nuvem, mas não mudou a senha em outro dispositivo. O outro dispositivo pode normalmente comunicar com o serviço de login com a senha anterior, o que pode causar problemas. Para evitar este problema, a senha de texto puro provida pelo usuário pode ser comparada contra o(s) blob(s) armazenado(s) como "senha corrente" e se nenhum coincidir, comparada contra o(s) blob(s) armazenado(s) como "senha anterior". Qualquer número desejado de conjuntos anteriores de um ou mais blob(s) de senha pode ser mantido exemplo, a senha corrente mais as duas últimas senhas podem também funcionar, e assim por diante.

[00052] Ainda, restrições de histórico de senha podem ser impostas

com o hash em repouso por exemplo, para usuários que não mudam as suas senhas local. Por exemplo, considere uma política onde os usuários não são permitidos reutilizar qualquer uma de suas cinco senhas anteriores. O serviço de nuvem armazena o último blob de senha, tal como (H3(Ha(password_current))), assim como um histórico de senha, tal como (H3(Ha(password_previous))); (H1(Ha(password _previous2))); H2(H1(Ha(password_previous3))) e assim por diante até o limite da política. Note que estes blobs não precisam ter sido gerados com o mesmo algoritmo de hash. Realmente, alguns destes podem ter sido novamente executado o hash se o hash original foi descoberto ser inseguro ou foi de outro modo mudado.

[00053] No tempo de mudança de senha, quando a nova senha é coletada, o serviço consulta a lista de algoritmos no campo de histórico de senha, computa os hashes correspondentes, e compara-os com as sinopses armazenadas para determinar se a mudança é permitida.

AMBIENTES EM REDE E DISTRIBUÍDOS EXEMPLARES

[00054] Alguém versado na técnica pode apreciar que as várias modalidades e métodos aqui descritos podem ser implementados em conexão com qualquer computador ou outro dispositivo de cliente ou de servidor, o qual pode ser posicionado como parte de uma rede de computador ou em um ambiente de computação distribuído, e pode ser conectado a qualquer tipo de armazenamento ou armazenamentos de dados. Neste aspecto, as várias modalidades aqui descritas podem ser implementadas em qualquer sistema de computador ou ambiente que tem qualquer número de memórias ou unidades de armazenamento, e qualquer número de aplicações e processos que ocorrem através de qualquer número de unidades de armazenamento. Isto inclui, mas não está limitado, um ambiente com computadores de servidor e computadores de cliente posicionados em um ambiente de rede ou um ambiente de computação distribuído, que tem armazenamento

remoto ou local.

[00055] A computação distribuída provê um compartilhamento recursos e serviços de computador por troca comunicativa entre os dispositivo e sistemas de computação. Estes recursos e serviços incluem a troca de informações, armazenamento em cache e armazenamento de disco para objetos, tal como arquivos. Estes recursos e serviços também incluem o compartilhamento de potência de processamento através de múltiplas unidades de processamento para balanceamento de carga, expansão de recursos, especialização de processamento, e similares. A computação distribuída se aproveita da conectividade de rede, permitindo os clientes alavancarem a sua potência coletiva para beneficiar a empresa inteira. Neste aspecto, uma variedade de dispositivos podem ter aplicações, objetos ou recursos que podem participar nos mecanismos de gerenciamento de recursos como descrito para várias modalidades da presente descrição.

[00056] A Figura 7 provê um diagrama esquemático de um ambiente de computação em rede ou distribuído exemplar. O ambiente de computação distribuído compreende os objetos de computação 710, 712, etc., e os objetos ou dispositivos de computação 720, 722, 724, 726, 728, etc., os quais podem incluir programas, métodos, armazenamento de dados, lógica programável, etc. como representado por aplicações exemplares 730, 732, 734, 736, 738. Pode ser apreciado que os objetos de computação 710, 712, etc. e os objetos ou dispositivos de computação 720, 722, 724, 726, 728, etc. podem compreender diferentes dispositivos, tal com assistentes digitais pessoais (PDAs), dispositivos de áudio/vídeo, telefones móveis, MP3 players, computadores pessoais, laptops, etc.

[00057] Cada objeto de computação 710, 712, etc. e objeto ou dispositivos de computação 720, 722, 724, 726, 728, etc. podem comunicar com um ou mais outros objetos de computação 710, 712, etc. e

objeto ou dispositivos de computação 720, 722, 724, 726, 728, etc. por meio da rede de comunicações 740, ou diretamente ou indiretamente. Apesar de ilustrada como um único elemento na Figura 7, rede de comunicações 740 pode compreender outros objetos de computação e dispositivos de computação que proveem serviços para o sistema da Figura 7, e/ou podem representar múltiplas redes interconectadas, as quais não estão mostradas. Cada objeto de computação 710, 712, etc. ou objeto de computação ou dispositivo 720, 722, 724, 726, 728, etc. pode também conter uma aplicação, tal como as aplicações 730, 732, 734, 736, 738, que poderiam fazer uso de uma API, ou outro objeto, software, firmware e/ou hardware, adequado para comunicação com ou implementação da aplicação provida de acordo com várias modalidades da presente descrição.

[00058] Existe uma variedade de sistemas, componentes, e configurações de rede que suportam os ambientes de computação distribuídos. Por exemplo, os sistemas de computação podem ser conectados juntos por sistemas com fio ou sem fio, por redes locais ou redes amplamente distribuídas. Correntemente, muitas redes estão acopladas na Internet, o que provê uma infraestrutura para computação amplamente distribuída e abrange muitas diferentes redes, apesar de que qualquer infraestrutura de rede pode ser utilizada para as comunicações exemplares feitas incidentes aos sistemas como descrito em várias modalidades.

[00059] Assim, um hospedeiro de topologias de rede e infraestruturas rede, tal como arquiteturas cliente/servidor, ponto a ponto, ou híbridas, podem ser utilizadas. O "cliente" é um membro de uma classe ou grupo que utiliza os serviços de outra classe ou grupo com o qual este não está relacionado. Um cliente pode ser um processo, por exemplo, aproximadamente um conjunto de instruções ou tarefas, que solicita um serviço provido por outro programa ou processo. O proces-

so de cliente utiliza o serviço solicitado sem precisar "conhecer" qualquer detalhe de trabalho sobre o outro programa ou o próprio serviço.

[00060] Em uma arquitetura de cliente/servidor, especificamente um sistema em rede, um cliente é usualmente um computador que acessa recursos de rede compartilhados providos por outro computador, por exemplo, um servidor. Na ilustração da Figura 7, como um exemplo não limitante, os objeto ou dispositivos de computação 720, 722, 724, 726, 728, etc. podem ser imaginados como clientes e os objetos de computação 710, 712, etc. podem ser imaginados como servidores onde os objetos de computação 710, 712, etc., atuando como servidores proveem serviços de dados, tal como recebendo dados de objetos ou dispositivos de computação de cliente 720, 722, 724, 726, 728, etc., armazenamento de dados, processamento de dados, transmissão de dados para objetos ou dispositivos de computação de cliente 720, 722, 724, 726, 728, etc., apesar de que qualquer computador pode ser considerado um, um servidor, ou ambos, dependendo das circunstâncias.

[00061] Um servidor é tipicamente um sistema de computador remoto acessível sobre uma rede remota ou local, tal como Internet ou infraestruturas de rede sem fio. O processo de cliente pode estar ativo em um primeiro sistema de computador, e o servidor processo pode estar ativo em um segundo sistema de computador, comunicando um com o outro sobre um meio de comunicações, assim provendo uma funcionalidade distribuída e permitindo que múltiplos clientes se aproveitem da capacidade de acúmulo de informações do servidor.

[00062] Em um ambiente de rede no qual a rede de comunicações 740 ou barramento é a Internet, por exemplo, os objetos de computação 710, 712, etc. podem ser servidores da Web com os quais outros objetos ou dispositivos de computação 720, 722, 724, 726, 728, etc. comunicam através de qualquer de um número de protocolos conhecidos, tal como o protocolo de transferência de hipertexto (HTTP). Os

objetos de computação 710, 712, etc. que atuam como servidores podem também servir como clientes, por exemplo, objetos ou dispositivos de computação 720, 722, 724, 726, 728, etc., como pode ser característico de um ambiente de computação distribuído.

DISPOSITIVO DE COMPUTAÇÃO EXEMPLAR

[00063] Como mencionado, vantajosamente, as técnicas aqui descritas podem ser aplicadas a qualquer dispositivo. Pode ser compreendido, portanto, que dispositivos de mão portáteis e outros dispositivos de computação e objetos de computação de todos os tipos são contemplados para utilização em conexão com as várias modalidades. Consequentemente, o computador remoto de uso geral abaixo, abaixo descrito na Figura 8 é apenas um exemplo de um dispositivo de computação.

[00064] As modalidades podem parcialmente ser implementadas através de um sistema de operação, para utilização por um desenvolvedor de serviços para um dispositivo ou objeto, e/ou incluídas dentro de software de aplicação que opera para desempenhar um ou mais aspectos funcionais das várias modalidades aqui descritas. O software pode ser descrito no contexto geral de instruções executáveis por computador, tal como módulos de programa, sendo executados por um ou mais computadores, tal como estações de trabalho de cliente, servidores ou outros dispositivos. Aqueles versados na técnica apreciarão que os sistemas de computador têm uma variedade de configurações e protocolos que podem ser utilizados para comunicar dados, e assim, nenhuma configuração ou protocolo específico é considerado limitante.

[00065] A Figura 8 assim ilustra um exemplo de um ambiente de sistema de computação adequado 800 no qual um ou mais aspectos das modalidades aqui descritas podem ser implementados, porém como tornado claro acima, o ambiente de sistema de computação 800

é somente um exemplo de um ambiente de computação adequado e não pretende sugerir nenhuma limitação quanto ao escopo de utilização ou funcionalidade. Além disso, o ambiente de sistema de computação 800 não pretende ser interpretado como tendo qualquer dependência em relação a qualquer um ou combinação de componentes ilustrados no ambiente de sistema de computação exemplar 800.

[00066] Com referência à Figura 8, um dispositivo remoto exemplar para implementar uma ou mais modalidades inclui um dispositivo de computação de uso geral na forma de um computador 810. Os componentes do computador 810 podem incluir, mas não estão limitados a unidade de processamento 820, uma memória de sistema 830, e a barramento de sistemas 822 que acopla vários componentes de sistema incluindo a memória de sistema na unidade de processamento 820.

[00067] O computador 810 tipicamente inclui uma variedade de meios legíveis por computador e pode ser qualquer meio disponível que possa ser acessado pelo computador 810. A memória de sistema 830 pode incluir um meio de armazenamento de computador na forma de uma memória volátil e/ou não volátil tal como uma memória somente de leitura (ROM) e/ou memória de acesso randômico (RAM). Como exemplo, e não limitação. a memória de sistema 830 pode também incluir um sistema de operação, programas de aplicação, outros módulos de programa, e dados de programa.

[00068] Um usuário pode inserir comandos e informações no computador 810 através de dispositivos de entrada 840. Um monitor ou outro tipo de dispositivo de exibição está também conectado no barramento de sistemas 822 através de uma interface, tal como a interface de saída 850. Além de um monitor, computadores podem também incluir outros dispositivos de saída periféricos tal como alto-falantes e uma impressora, a qual pode estar conectada através da interface de

saída 850.

[00069] O computador 810 pode operar em um ambiente em rede ou distribuído utilizando conexões lógicas para um ou mais outros computadores remotos, tal como o computador remoto 870. O computador remoto 870 pode ser um computador pessoal, um servidor, um roteador, um PC de rede, um dispositivo de ponto ou outro nodo de rede comum, ou qualquer outro dispositivo de consumo ou transmissão de mídia remoto, e pode incluir qualquer ou todos os elementos acima descrito em relação ao computador 810. As conexões lógicas apresentadas na Figura 8 incluem uma rede 872, tal como uma rede de área local (LAN) ou uma rede de área ampla (WAN), mas pode também incluir outras redes/barramentos. Tais ambientes de rede são comuns em residências, escritórios, redes de computador de amplitude de empresa, intranets e a Internet.

[00070] Como acima mencionado apesar de modalidades exemplares terem sido descritas em conexão com vários dispositivo de computação e arquiteturas de rede, os conceitos subjacentes podem ser aplicados a qualquer sistema de rede e qualquer dispositivo ou sistema de computação no qual é desejável aperfeiçoar a eficiência de utilização de recursos.

[00071] Também, existem múltiplos modos para implementar a mesma ou similar funcionalidade, por exemplo, uma API apropriada, kit de ferramentas, código de driver, sistema de operação, controle, objeto de software independente ou capaz de ser feito download, etc. o que permite que as aplicações e serviços se aproveitem das técnicas aqui providas. Assim, as modalidades aqui são contempladas do ponto de vista de uma API (ou outro objeto de software), assim como de um objeto de software ou hardware que implementa uma ou mais modalidades como aqui descrito. Assim, várias modalidades aqui descritas podem ter aspectos que são totalmente em hardware, parcialmente

em hardware e parcialmente em software, assim como em software.

[00072] A palavra "exemplar" é aqui utilizada para significar servindo como um exemplo, instância, ou ilustração. Para evitar dúvidas, o assunto aqui descrito não está limitado por tais exemplos. Além disso, qualquer aspecto ou projeto aqui descrito como "exemplar" não deve necessariamente ser considerado como preferido ou vantajoso sobre outros aspectos ou projetos, nem pretende excluir estruturas e técnicas exemplares equivalentes conhecidas daqueles versados na técnica. Mais ainda, no grua em que os termos "inclui", "tem", "contém" e outras palavras similares são utilizadas, para evitar dúvida tais termos são pretendidos serem inclusivos em um modo similar ao termo "compreendendo" como uma palavra de transição aberta sem excluir nenhum elemento adicional ou outro quando empregada em uma reivindicação.

[00073] Como mencionado, as várias aqui descritas podem ser implementadas em conexão com hardware ou software ou, onde apropriado com uma combinação de ambos. Como aqui utilizado, os termos "componente", "módulo", "sistema" e similares são do mesmo modo pretendidos referir a uma entidade relativa a computador ou hardware, uma combinação de hardware e software, software, ou software em execução. Por exemplo, um componente pode ser, mas não está limitado a ser, um processo executando em um processor, um processor, um objeto, um executável, uma cadeia de execução, um programa, e/ou um computador. Como ilustração, tanto uma aplicação que executa no computador quanto o computador pode ser um componente. Um ou mais componentes podem residir dentro de um processo e/ou cadeia de execução e um componente pode estar localizado em um computador e/ou distribuído entre dois ou mais computadores.

[00074] Os sistemas acima mencionados foram descritos com rela-

ção à interação entre diversos componentes. Pode ser apreciado que tais sistemas e componentes podem incluir aqueles componentes ou subcomponentes especificados, alguns dos componentes ou subcomponentes especificados, e/ou componentes adicionais, e de acordo com várias permutações e combinações dos acima. Os subcomponentes podem também ser implementados como componentes comunicativamente acoplados a outros componentes ao invés de incluídos dentro de componentes pais (hierárquico). Além disso, pode ser notado que um ou mais componentes podem ser combinados em um único componente provendo uma funcionalidade agregada ou dividido em diversos subcomponentes separados, e que qualquer uma ou mais camadas intermediárias, tal como uma camada de gerenciamento, podem ser providas para comunicativamente acoplar a tais subcomponentes de modo a prover uma funcionalidade integrada. Quaisquer componentes aqui descritos podem também interagir com um ou mais outros componentes não especificamente aqui descritos mas geralmente conhecidos daqueles versados na técnica.

[00075] Em vista dos sistemas exemplares aqui descritos, uma metodologia pode ser implementada de acordo com o assunto descrito podem ser apreciadas com referência aos fluxogramas das várias figuras. Apesar dos propósitos de simplicidade de explanação, as metodologias estão mostradas e descritas como uma série de blocos, deve ser compreendido e apreciado que as várias modalidades não estão limitadas pela ordem dos blocos, juá que alguns blocos podem ocorrer em diferentes ordens e/ou concorrentemente com outros blocos do que é aqui apresentado e descrito. Onde um fluxo não sequencial, ou ramificado, é ilustrado através de fluxograma, pode ser apreciado que várias outras ramificações, percursos de fluxo, e ordens dos blocos, podem ser implementados os quais atingem o mesmo ou um similar resultado. Mais ainda, alguns blocos ilustrados são opcionais na im-

plementação da metodologias aqui acima descritas.

CONCLUSÃO

[00076] Apesar da invenção ser suscetível a várias modificações e construções alternativas, certas suas modalidades ilustradas estão mostradas nos desenhos e foram acima descritas em detalhes. Deve ser compreendido, no entanto, que não há intenção em limitar a invenção às formas específicas descritas, mas ao contrário, a intenção é cobrir todas as modificações, construções alternativas, e equivalentes que caem dentro do espírito e escopo da invenção.

[00077] Além das várias modalidades aqui descritas, deve ser compreendido que outras modalidades similares podem ser utilizadas ou modificações e adições podem ser feitas na(s) modalidade(s) descritas para desempenhar as mesmas funções ou equivalentes que da(s) modalidade(s) correspondentes sem desviar destas. Mais ainda, múltiplos chips de processamento ou múltiplos dispositivos podem compartilhar o desempenho de uma ou mais funções aqui descritas, e similarmente, o armazenamento pode ser efetuado através de uma pluralidade de dispositivos. Consequentemente, a invenção não está limitada a qualquer única modalidade, mas ao invés deve ser considerada em amplitude, espírito e escopo de acordo com as reivindicações anexas.

REIVINDICAÇÕES

1. Método para sincronizar um conjunto de credenciais de senha entre um serviço de origem e um serviço alvo, o conjunto de credenciais de senha permitindo acesso a cada um do serviço de origem e do serviço alvo, o método **caracterizado pelo fato de que** compreende as etapas de:

proporcionar, por um processador, uma estampa de tempo de sincronização que corresponde a uma última sincronização de credencial de senha entre o serviço de origem e o serviço alvo;

receber um conjunto de credenciais de senha mudadas em hash, o conjunto de credenciais de senha mudadas em hash incluindo apenas mudanças feitas ao conjunto de credenciais de senha desde a última sincronização de credencial de senha; e

exportar o conjunto de credenciais de senha mudadas para o serviço alvo para utilização em autenticação de identidade.

2. Método, de acordo com a reivindicação 1, **caracterizado** pelo fato de que o conjunto de credenciais de senha mudadas em hash é computado com um algoritmo de hash primário, e em que o método ainda compreende secundariamente executar um hash no conjunto de credenciais de senha mudadas em hash em dados protegidos por segredo utilizando um algoritmo de hash secundário que corresponde ao conjunto de credenciais de senha mudadas em hash para exportar para o serviço alvo.

3. Método, de acordo com a reivindicação 2, **caracterizado** pelo fato de que secundariamente executar um hash no conjunto de credenciais de senha mudadas em hash nos dados protegidos por segredo compreende utilizar salt randômico e um número de iterações.

4. Método, de acordo com a reivindicação 1, **caracterizado** pelo fato de que receber o conjunto de credenciais de senha mudadas em hash compreende solicitar dados de mudança a partir de um servi-

ço de diretório.

5. Método, de acordo com a reivindicação 4, **caracterizado** pelo fato de que ainda compreende analisar os dados de mudança em dados de mudança de senha compreendendo o conjunto de credenciais de senha mudadas em hash.

6. Método, de acordo com a reivindicação 1, **caracterizado** pelo fato de que compreende ainda receber um valor de status em resposta a exportar o conjunto de credenciais de senha mudadas em hash, e se o valor de status não indicar sucesso, enfileirar o conjunto de credenciais de senha mudadas em hash para uma tentativa de exportação subsequente.

7. Método, de acordo com a reivindicação 1, **caracterizado** pelo fato de que ainda compreende:

receber o conjunto de credenciais de senha mudadas em hash no serviço alvo, incluindo receber informação que identifica qual um ou mais algoritmos de hash secundários utilizar quando se autentica uma identidade correspondente conjunto de credenciais de senha mudadas em hash.

8. Método, de acordo com a reivindicação 1, **caracterizado** pelo fato de que ainda compreende:

receber, pelo serviço alvo, o conjunto de credenciais de senha mudadas em hash; e

adicionalmente executar um hash do conjunto de credenciais de senha mudadas em hash com pelo menos uma função de hash adicional no serviço alvo para armazenar como dados protegidos por segredo.

9. Dispositivo de armazenamento legível por computador, **caracterizado pelo fato de que** possui um método que, quando executado por um ou mais processadores, faz com que um ou mais processadores executem:

manter, em um serviço alvo, uma pluralidade de conjuntos de dados compreendendo dados de senha protegidos correspondentes a senhas de texto sem formatação, os dados de senha protegidos associados à uma identidade respectiva, em que os dados de senha protegida são computados a partir de uma senha de texto sem formatação por uma combinação de pelo menos dois algoritmos de hash e sincronizados com o serviço alvo;

receber, no serviço alvo, uma tentativa de login, incluindo uma credencial correspondente a uma identidade e uma senha;

computar um primeiro valor com base na execução de pelo menos um algoritmo de hash na senha; e

comparar o primeiro valor com dados de senha protegidos associados à identidade para autenticar a identidade.

10. Dispositivo de armazenamento legível por computador, de acordo com a reivindicação 9, **caracterizado** pelo fato de que ainda compreende substituir os dados associados a uma identidade por dados computados a partir de pelo menos um algoritmo de hash diferente.

11. Dispositivo de armazenamento legível por computador, de acordo com a reivindicação 9, **caracterizado** pelo fato de que ainda compreende adicionar novos dados em associação com uma identidade, na qual os novos dados são calculados a partir de pelo menos um algoritmo de hash diferente.

12. Dispositivo de armazenamento legível por computador, de acordo com a reivindicação 9, **caracterizado** pelo fato de que ainda compreende manter os dados atuais e um histórico compreendendo pelo menos um conjunto anterior de dados em associação com uma identidade, e em que comparar o primeiro valor com os dados de senha protegidos associados à identidade para autenticar a identidade compreende o uso da história.

13. Dispositivo de armazenamento legível por computador, de acordo com a reivindicação 9, **caracterizado** pelo fato de que ainda compreende receber, no serviço de destino, uma indicação de que um serviço de diretório local está usando um algoritmo de hash primário diferente.

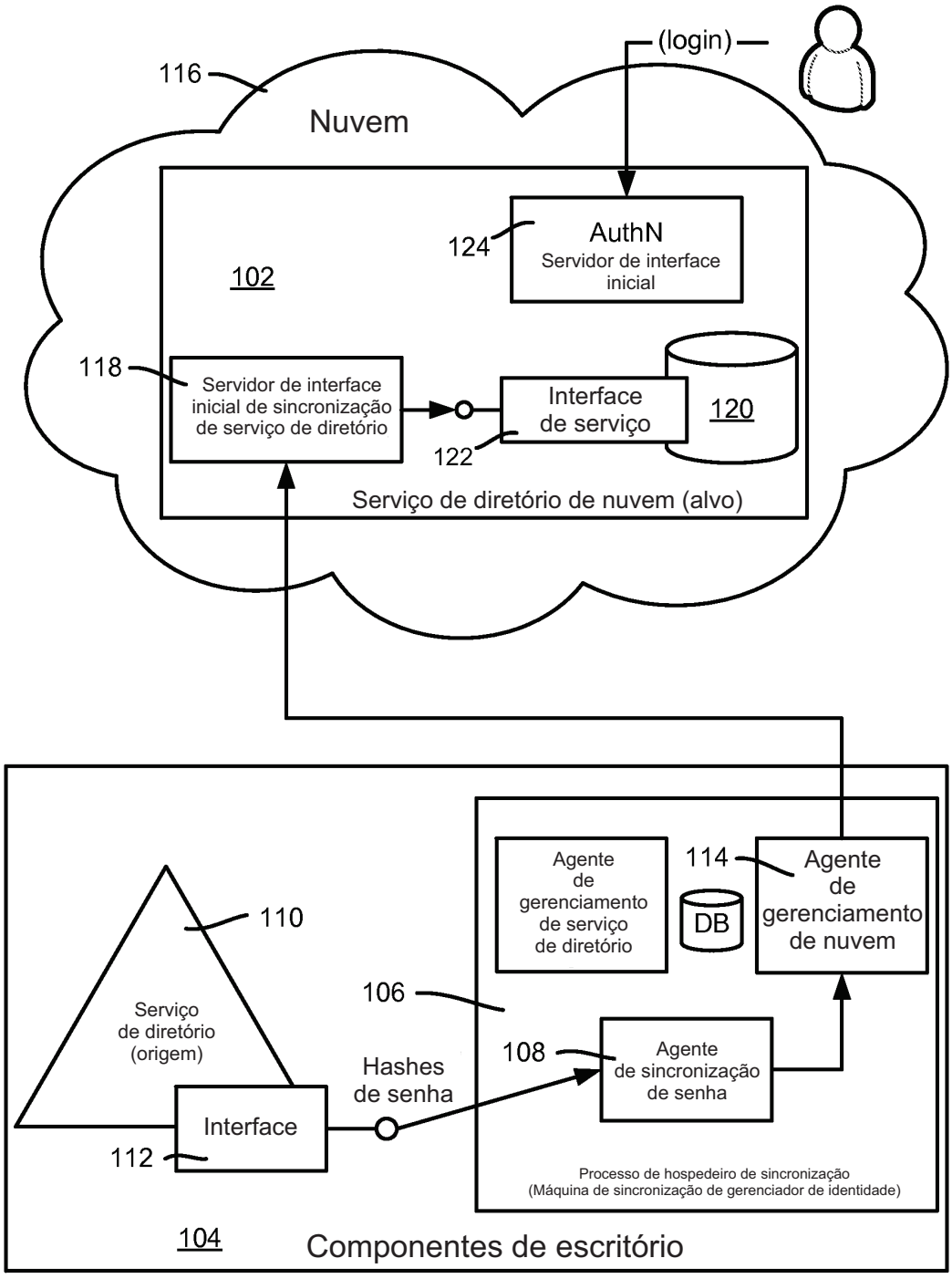
(login)
116
Nuvem
124
AuthN
Servidor de interface inicial
102
118
Servidor de interface inicial de sincronização de serviço de diretório
Interface de serviço
120
122
Serviço de diretório de nuvem (alvo)
Agente de gerenciamento de serviço de diretório
114
Agente de gerenciamento de nuvem
DB
110
Serviço de diretório (origem)
106
108
Agente de sincronização de senha
Hashes de senha
Interface
112
Processo de hospedeiro de sincronização
(Máquina de sincronização de gerenciador de identidade)
104
Componentes de escritório

FIG. 1

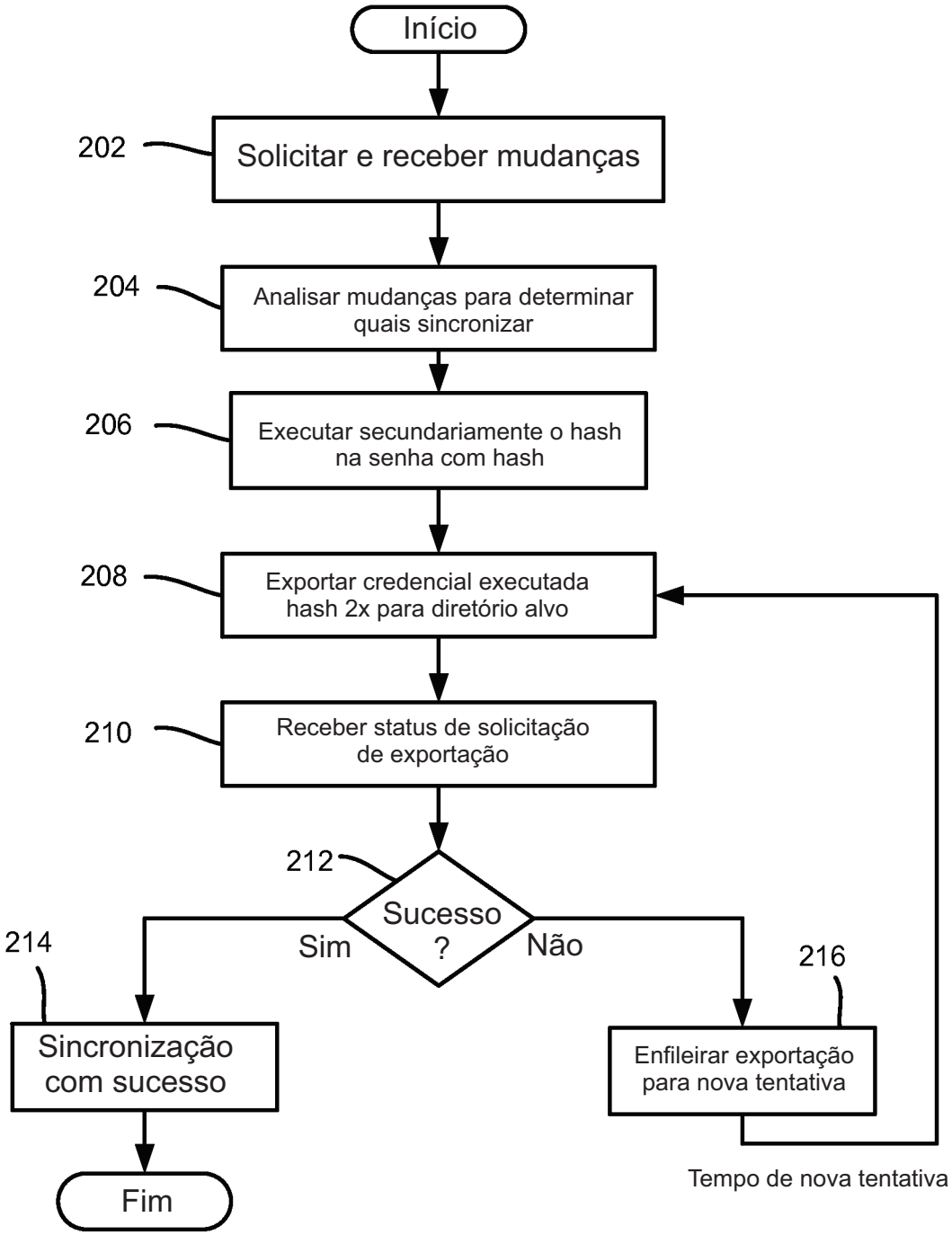
Início
202
Solicitar e receber mudanças
204
Analisar mudanças para determinar quais sincronizar
206
Executar secundariamente o hash na senha com hash
208
Exportar credencial executada hash 2x para diretório alvo
210
Receber status de solicitação de exportação
212
Sucesso ?
Sim
Não
214
Sincronização com sucesso
216
Enfileirar exportação para nova tentativa
Tempo de nova tentativa
Fim

FIG. 2

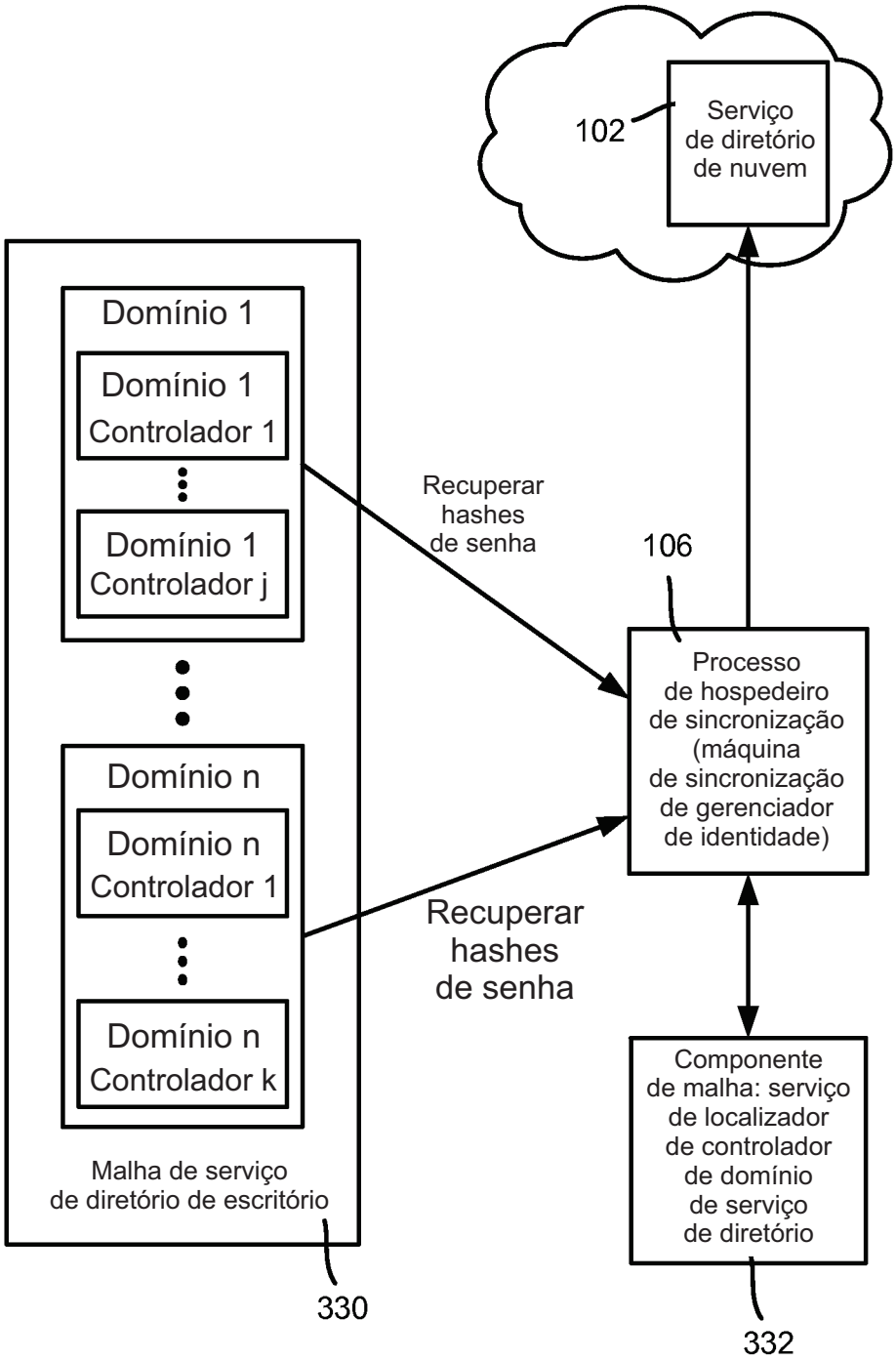
102
Serviço
de diretório
de nuvem
Domínio 1
Domínio 1
Controlador 1
Domínio 1
Controlador j
Domínio n
Domínio n
Controlador 1
Domínio n
Controlador k
Malha de serviço
de diretório de escritório
330
Recuperar
hashes
de senha
106
Processo
de hospedeiro
de sincronização
(máquina
de sincronização
de gerenciador
de identidade)
Recuperar
hashes
de senha
Componente
de malha: serviço
de localizador
de controlador
de domínio
de serviço
de diretório
332

FIG. 3

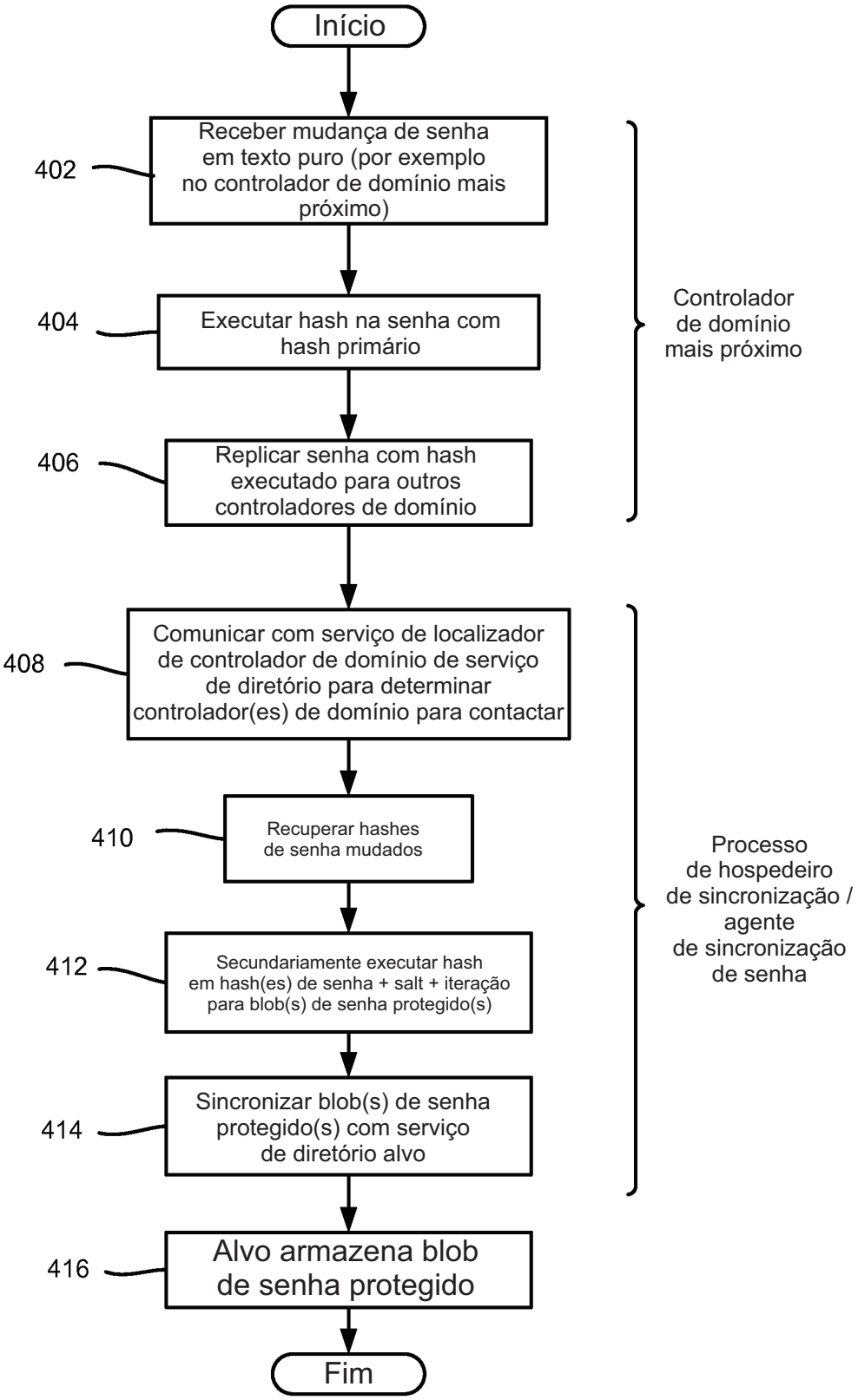
Início
Receber mudança de senha em texto puro (por exemplo no controlador de domínio mais próximo)
402
Executar hash na senha com hash primário
404
Replicar senha com hash executado para outros controladores de domínio
406
Controlador de domínio mais próximo
Comunicar com serviço de localizador de controlador de domínio de serviço de diretório para determinar controlador(es) de domínio para contactar
408
Recuperar hashes de senha mudados
410
Secundariamente executar hash em hash(es) de senha + salt + iteração para blob(s) de senha protegido(s)
412
Sincronizar blob(s) de senha protegido(s) com serviço de diretório alvo
414
Processo de hospedeiro de sincronização / agente de sincronização de senha
Alvo armazena blob de senha protegido
416
Fim

FIG. 4

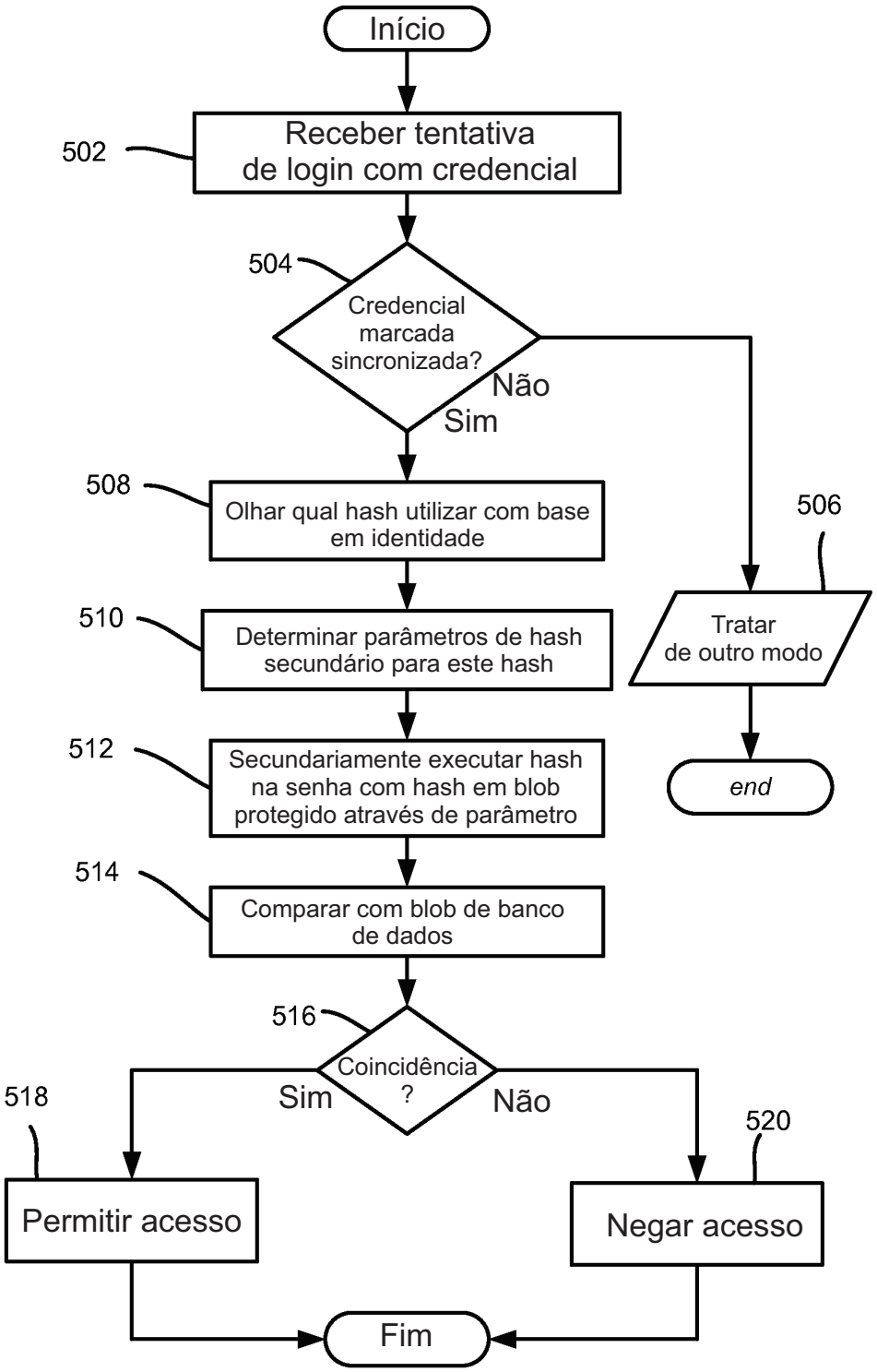
Início
Receber tentativa de login com credencial
502
504
Credencial marcada sincronizada?
Não
Sim
506
Tratar de outro modo
end
508
Olhar qual hash utilizar com base em identidade
510
Determinar parâmetros de hash secundário para este hash
512
Secundariamente executar hash na senha com hash em blob protegido através de parâmetro
514
Comparar com blob de banco de dados
516
Coincidência ?
Sim
Não
518
Permitir acesso
520
Negar acesso
Fim

FIG. 5

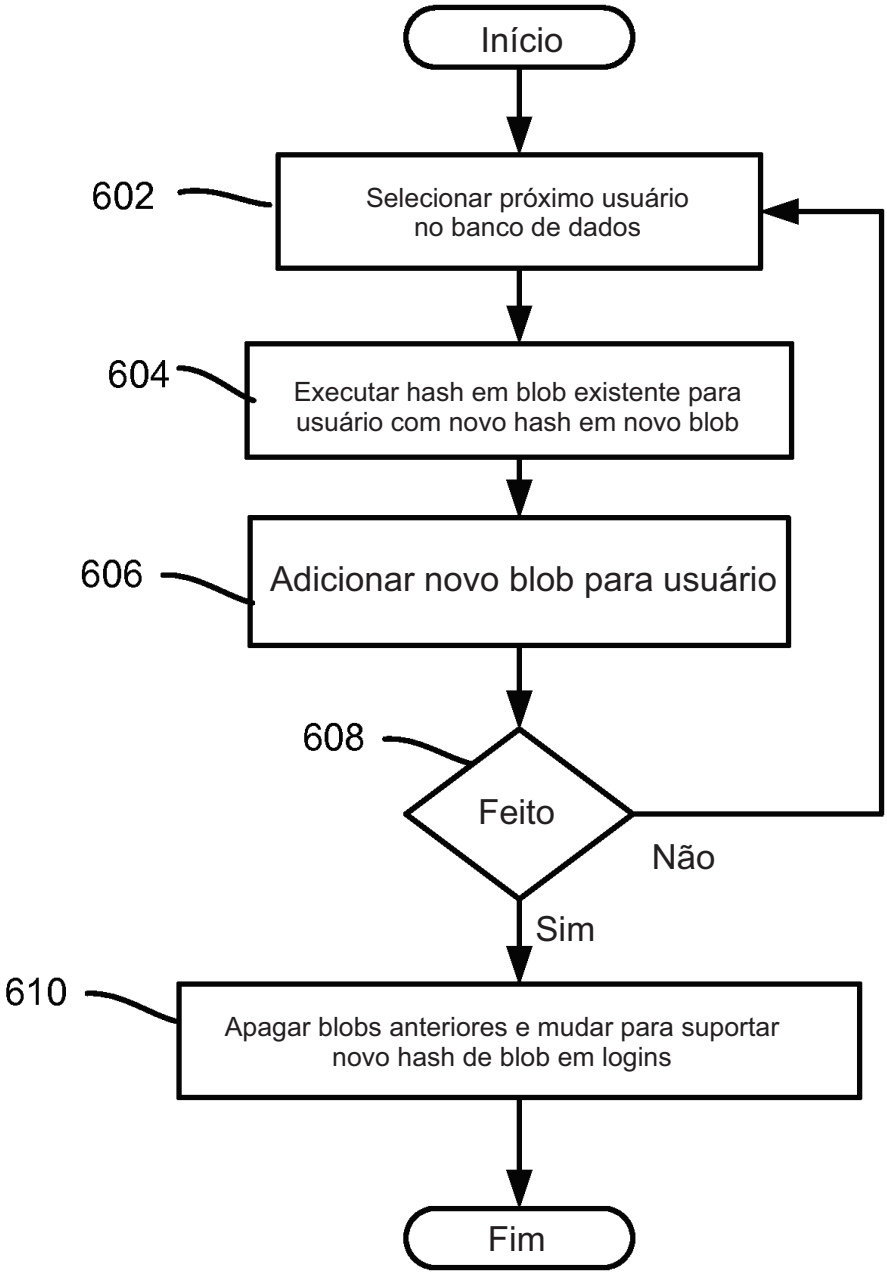
Início
602
Selecionar próximo usuário no banco de dados
604
Executar hash em blob existente para usuário com novo hash em novo blob
606
Adicionar novo blob para usuário
608
Feito
Não
Sim
610
Apagar blobs anteriores e mudar para suportar novo hash de blob em logins
Fim

FIG. 6

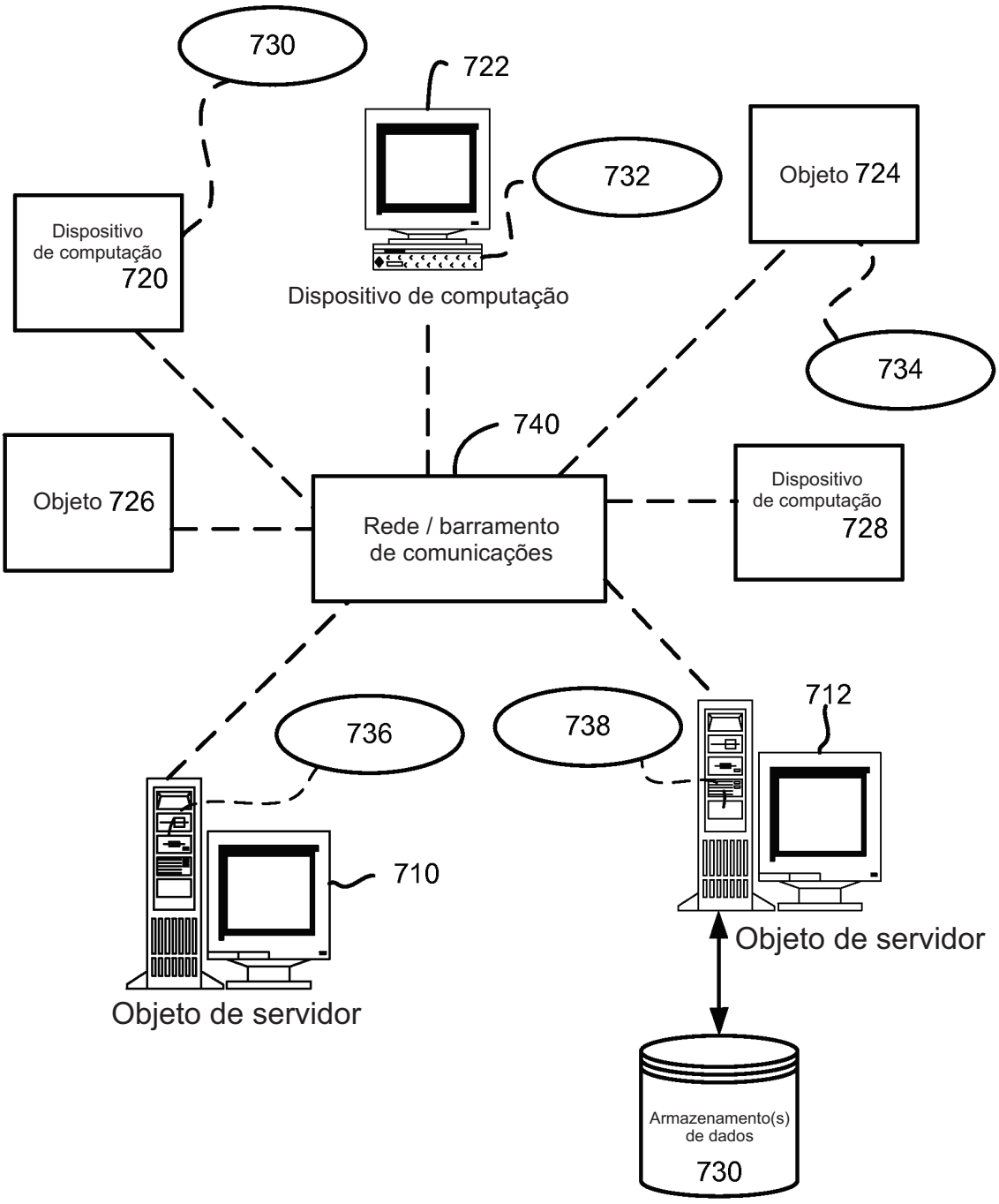
730
722
732
Objeto 724
Dispositivo de computação 720
Dispositivo de computação
734
740
Objeto 726
Rede / barramento de comunicações
Dispositivo de computação 728
736
738
712
710
Objeto de servidor
Objeto de servidor
Armazenamento(s) de dados 730

FIG. 7

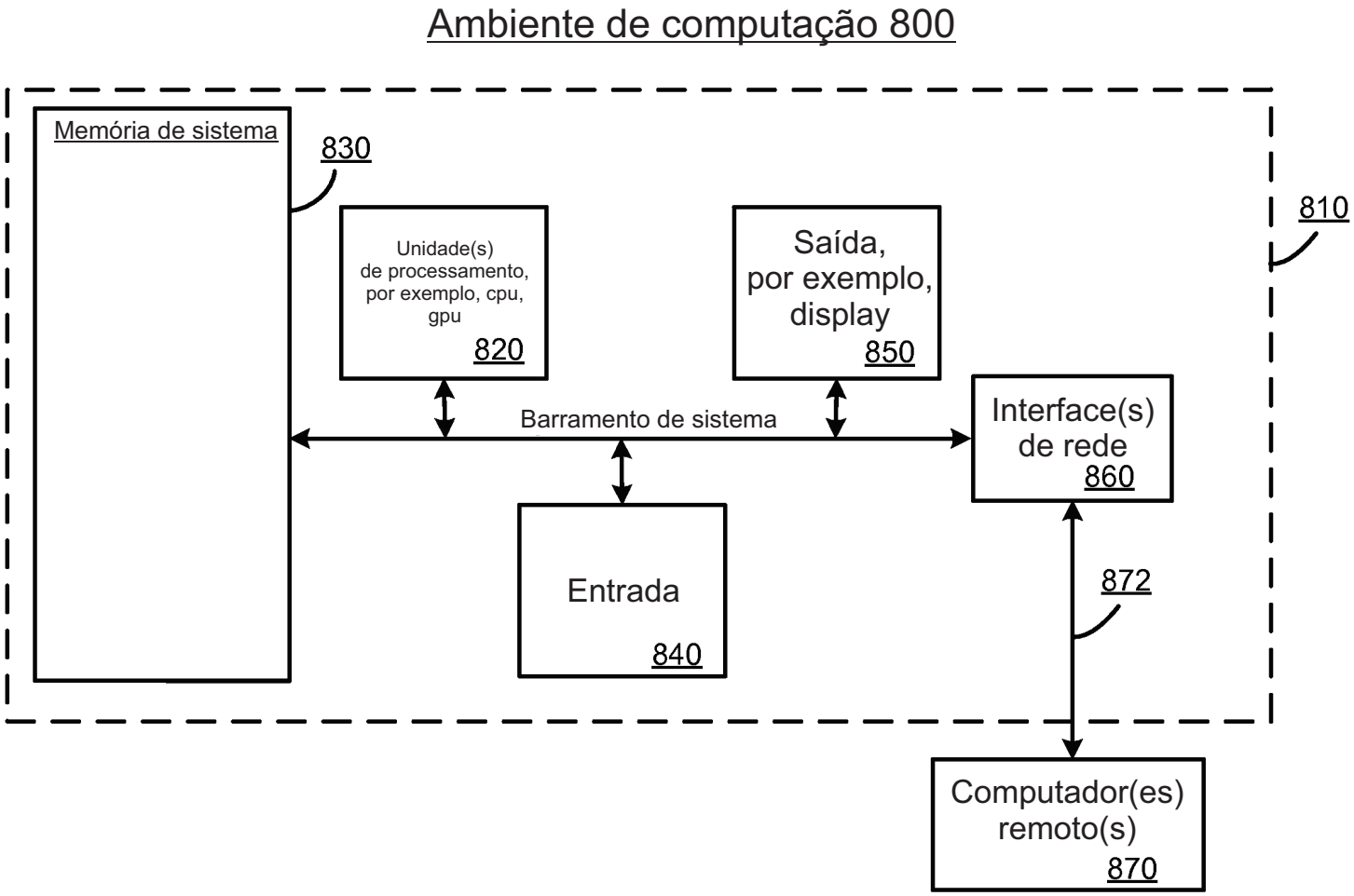
Ambiente de computação 800
Memória de sistema
830
Unidade(s) de processamento, por exemplo, cpu, gpu
820
Saída, por exemplo, display
850
810
Barramento de sistema
Interface(s) de rede
860
Entrada
840
872
Computador(es) remoto(s)
870

FIG. 8