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

(11) PI 0621674-9 B1

(22) Data do Depósito: 15/05/2006

(45) Data de Concessão: 28/05/2019

(54) Título: MÉTODO E SISTEMA PARA AUTENTICAÇÃO FORA DE BANDA DE FLUXOS DE DADOS TRANSMITIDOS ATRAVÉS DE UMA REDE DE COMUNICAÇÃO

(51) Int.Cl.: H04L 29/06.

(52) CPC: H04L 63/123; H04L 63/126; H04L 63/18.

(73) Titular(es): TELECOM ITALIA S.P.A.

(72) Inventor(es): PAOLO DE LUTUS; CORRADO MOISO; GAETANO DI CAPRIO.

(86) Pedido PCT: PCT EP2006004555 de 15/05/2006

(87) Publicação PCT: WO 2007/131523 de 22/11/2007

(85) Data do Início da Fase Nacional: 14/11/2008

(57) Resumo: MÉTODO E SISTEMA PARA AUTENTICAÇÃO FORA DE BANDA DE FLUXOS DE DADOS TRANSMITIDOS ATRAVÉS DE UMA REDE DE COMUNICAÇÃO. Um método e sistema para autenticação fora de banda de mensagens transmitidas, por exemplo como pacotes, em uma rede de comunicação (4), por meio de que um primeiro fluxo de dados é recebido por um módulo de controle de remetente (10) de um remetente (2); o primeiro fluxo de dados é transmitido através de primeiro canal, por exemplo um canal de dados não seguro (7), para um módulo de controle de receptor (11); o módulo de controle de remetente gera dados de autenticação do primeiro fluxo de dados; os dados de autenticação são transmitidos do módulo de controle de remetente (10) para o módulo de controle de receptor (11) em um segundo canal, por exemplo um canal de dados seguro (8), distinto do primeiro canal (7); e um fluxo de dados recebidos pelo módulo de controle de receptor (11) é verificado usando os dados de autenticação. Antes de enviar os dados de autenticação, o módulo de controle de remetente (10) transmite uma mensagem de controle incluindo dados de sincronização para o módulo de controle de receptor (11) através do segundo canal (8).

“MÉTODO E SISTEMA PARA AUTENTICAÇÃO FORA DE BANDA DE FLUXOS DE DADOS TRANSMITIDOS ATRAVÉS DE UMA REDE DE COMUNICAÇÃO”

CAMPO TÉCNICO DA INVENÇÃO

[0001] A presente invenção relaciona-se em geral ao campo de redes de comunicação, e em particular a um método de autenticação fora de banda para transmissão em fluxo e comunicação de mensagem discreta por uma rede de dados. Mais particularmente, a invenção relaciona-se a uma solução assegurando a integridade dos dados e a autenticidade das partes de uma conexão para troca de dados usando uma rede de comunicação pública do tipo de pacote, por exemplo uma rede de Protocolo da Internet (IP).

FUNDAMENTOS DA TÉCNICA

[0002] Como é conhecido, o uso de sistemas implementando redes de comunicação públicas como pacote, como a rede de IP, para troca de dados e/ou transmissão em fluxo de multimídia requer o uso de soluções de segurança capazes de assegurar a integridade dos dados e a autenticidade das partes em uma conexão.

[0003] As soluções mais usadas para implementar tais medidas de segurança provêem geralmente uma extensão dos protocolos de comunicação existentes em virtude da introdução de novas porções (por exemplo campos específicos) ou a modificação de campos já existentes dentro de protocolos de aplicativo.

[0004] Em geral, todo protocolo de aplicativo (tais como Protocolo de Acesso de Objeto simples - SOAP; Protocolo de Transferência de hipertexto - HTTP; Invocação de Método Remoto de Java - RMI; Protocolo Inter-orbe da Internet - IIOP) define um ou mais campos cujo conteúdo está definido de acordo com o protocolo específico. Por exemplo, SOAP, usado na invocação de serviços da web, inclui um "cabeçalho" e um "corpo", usados para descrever o conteúdo da mensagem transmitida. Uma extensão "segura" de

SOAP, chamada Segurança-WS, introduz campos específicos ambos para transmitir informação sobre a identidade do aplicativo invocador (por exemplo, no cabeçalho de SOAP) e para assegurar a integridade do pedido ou a resposta de invocação de serviço da web (contida no corpo de SOAP).

[0005] A abordagem anterior não pode ser sempre aplicada desde que não é sempre possível modificar ou estender os protocolos existentes ou os aplicativos usando-os (aplicativos de "legado") assim para adicionar informação de autenticação e integridade às mensagens transmitidas. Exemplos de protocolos que não podem ser modificados são RTP (Protocolo em Tempo Real), FTP (Protocolo de Transferência de Arquivo), Protocolo de Telnet e muitos outros protocolos do grupo Protocolo de Controle de transmissão - TCP/IP/Protocolo de Internet, desde que estes protocolos foram projetados sem levar em conta quaisquer requisitos de segurança.

[0006] Outras soluções para implementar medidas de segurança incluem usar protocolos de aplicativo que são "seguros" a um nível de transporte ou de rede (de acordo com a Interconexão de Sistema Aberto (OSI) - modelo de ISO/OSI da Organização de Padrões Internacionais (ISO)). Exemplos de tais soluções são SSL (Camada de Receptáculo Seguro) ou protocolos provendo túneis seguros, tal como IPSEC (IPSecurity). Esta abordagem provê geralmente codificação e autenticação de mensagem encapsulando o fluxo de comunicação original em mensagens do protocolo de comunicação seguro.

[0007] US 6.842.860 expõe uma solução usando um código de autenticação de mensagem parcial, em que um código de autenticação de mensagem é aplicado só a algumas porções da mensagem.

[0008] US 2005/0228983 expõe um sistema incluindo um canal lateral seguro e um canal de legado inseguro. Em uma concretização, um cliente utiliza a função *hash* em algum do conteúdo enviado através do canal inseguro e envia o *hash* através do canal seguro. O servidor então utiliza a

função *hash* no conteúdo recebido através do canal inseguro e compara o *hash* que gera ao recebido através do canal seguro para determinar se a mensagem postada através do canal inseguro foi alterada.

[0009] US 200370120924 expõe um método para verificar a integridade de uma mensagem transmitida entre um remetente e um recebedor. Na ponta transmitida, um valor de autenticação é gerado de uma mensagem a ser enviada. Um código de verificação é formado do valor de autenticação e uma carreira aleatória. A primeira mensagem é transferida do remetente ao recebedor por um primeiro canal, e o código de verificação é transferido por um segundo canal seguro. Na ponta de recepção, uma verificação de autenticação é formada baseada na mensagem recebida. A integridade da mensagem recebida é verificada comparando os valores de verificação na ponta de recepção.

OBJETIVO E SUMÁRIO DA INVENÇÃO

[00010] O Requerente observou que soluções usando mecanismos de segurança operando a nível de transporte ou rede provêem uma proteção de ponta a ponta, mas são menos adequadas quando mensagens transmitidas deveriam passar por uma pluralidade de nós intermediários que precisam acessar a informação transmitida. Além disso, tais soluções são desvantajosas quando é importante que nenhum atraso seja introduzido na transmissão de fluxo de informação, tal como em comunicações em tempo real, como Voz através de IP, desde que qualquer atraso poderia deteriorar a qualidade de serviço.

[00011] Além disso, métodos de autenticação incluindo uma verificação off-line sobre a autenticidade das mensagens, não pode ser estendida a tráfego em tempo real ou transmissão em fluxo, em que mensagens geradas dinamicamente são enviadas continuamente (tal como em VoiceIP).

[00012] O Requerente também observou que soluções usando

mecanismos de código de autenticação de mensagem aplicados às mensagens transmitidas frequentemente requerem uma modificação dos aplicativos de cliente e servidor para administrar as mensagens modificadas, que não é sempre possível ou desejável.

[00013] O objetivo da presente invenção é, portanto, prover um método e um sistema para autenticação de mensagem que seja capaz de superar as desvantagens anteriores das soluções conhecidas.

[00014] Mídia digital pode ser transmitida através de uma rede em um fluxo contínuo, um método de entrega de conteúdo conhecido como transmissão em fluxo. O processo de transmissão em fluxo começa quando um arquivo de mídia é dividido em pedaços menores que assim podem ser transferidos e executados quando cada um dos pedaços é recebido, em lugar de esperar pelo arquivo inteiro ser transferido antes que reprodução comece. Em geral, fluxo de dados se refere a uma sequência de sinais codificados digitalmente usados para representar informação em transmissão.

[00015] Mídia contínua, tal como áudio ou vídeo em tempo real (por exemplo, estações de Rádio ou TV da Internet), é geralmente transferida (por exemplo, carregada) e executada usando tecnologias de transmissão em fluxo. Especialmente no caso de transmissão de fluxo contínua, o comprimento do fluxo de dados levando a informação pode ser *a priori* desconhecido ou, para os propósitos da administração da transmissão em fluxo, considerado infinito.

[00016] O Requerente entendeu que, especialmente enquanto lidando com transmissão em fluxo contínua, um mecanismo de autenticação deveria habilitar sincronização entre o remetente dos dados de informação e o receptor de forma que o receptor possa definir os dados de quais começar a verificação até mesmo se o tempo ao qual a transmissão de dados começou for desconhecido ao receptor.

[00017] De acordo com a presente invenção, é provido um método e um sistema para autenticação fora de banda de mensagens transmitidas por

uma rede de comunicação, como definido nas reivindicações 1 e 14, respectivamente.

[00018] Em particular, o método usa dois canais diferentes: um primeiro, que pode ser baseado em uma rede de comunicação do tipo de pacote com características seguras limitadas, por exemplo, Internet, para transmitir dados e/ou fluxos de dados (por exemplo, multimídia e dados em tempo real); e um segundo, que é preferivelmente um canal seguro, para enviar informação utilizável para verificar a integridade e/ou a autenticidade dos dados recebidos. Em particular, o método de autenticação pode ser usado para verificar integridade de dados, isto é, para verificar que os dados enviados são os mesmos dados que são recebidos, e/ou pode ser usado para a autenticação de origem, isto é, para verificar que os dados eram enviados de fato pelo remetente reivindicado. A fim de operar corretamente em fluxos de dados, dados de sincronização são trocados entre um módulo de remetente e um módulo de receptor. Assim, os módulos de remetente e receptor podem ser sincronizados, e o módulo de receptor pode executar as operações de verificação em tempo real.

BREVE DESCRIÇÃO DOS DESENHOS

[00019] Para um entendimento melhor da presente invenção, concretizações preferidas, que são planejadas puramente como exemplos e não são para serem interpretadas como limitantes, serão descritas agora com referência aos desenhos anexos, em que:

Figura 1 mostra um diagrama de bloco de um sistema de autenticação de acordo com uma primeira concretização da presente invenção;

Figura 2 mostra um diagrama de bloco de um sistema de autenticação de acordo com uma segunda concretização da presente invenção;

Figura 3 é um diagrama de tempo de uma transmissão entre um remetente e um receptor;

Figuras 4a, 4b e 4c são fluxogramas de uma concretização do método de autenticação de acordo com a presente invenção; e

Figura 5 descreve um possível fluxo entre um remetente e um receptor, de acordo com uma concretização do presente método.

DESCRIÇÃO DETALHADA DE CONCRETIZAÇÕES PREFERIDAS DA INVENÇÃO

[00020] Figura 1 mostra um sistema 1 para autenticar dados transmitidos por um remetente 2 para um receptor 3 por uma rede pública 4, de acordo com uma primeira concretização da presente invenção. Na concretização da Figura 1 ("solução de borda"), o remetente 2 e o receptor 3 são dois nós de comunicação dentro de uma LAN (Rede local) 5, 6, respectivamente. Por exemplo, o remetente 2 e o receptor 3 podem estar incluídos em uma rede de extranet empresarial formada por LANs confiadas diferentes, controladas por uma única unidade de controle central e conectadas entre si por uma rede de área extensa (WAN), por exemplo, a Internet.

[00021] Em particular, o remetente 2 e o receptor 3 podem ser por exemplo, um computador, um PDA (Assistente Digital Pessoal), um laptop, um computador portátil com capacidade sem fios, um telefone de VoIP, uma câmera da Web ou um telefone de IP sem fios.

[00022] A rede pública 4, por exemplo a Internet, inclui um primeiro canal 7 e um segundo canal 8. Primeiro e segundo canais 7 e 8 são canais de dados lógicos não correspondendo necessariamente a canais físicos diferentes desde que um canal físico pode ser compartilhado por mais canais lógicos. Exemplos de canais físicos podem ser cabos ou canais de rádio no caso de comunicação móvel, tais como Canais de Dados de Pacote (PDCH) em GPRS, cada canal estando associado com um intervalo de tempo de um quadro de TDMA.

[00023] Primeiro canal 7 é usado para transmitir dados,

preferivelmente dados de pacote, tanto como mensagens discretas ou como fluxos de dados (por exemplo, multimídia e dados em tempo real) e tem características de segurança não especificadas (por exemplo, limitadas). Segundo canal 8 é preferivelmente um canal seguro e é implementado de qualquer modo conhecido para transmitir dados de controle de um modo seguro. Protocolos de segurança que operam a nível de aplicativo, de transporte ou de rede, por exemplo podem ser implementados por SSL (Camada de Receptáculo Seguro) ou IPSEC (IPSecurity) pode ser empregado no segundo canal 8.

[00024] Cada nó de comunicação 5 e 6 inclui um módulo de controle 10 e 11, respectivamente, conectado ao remetente 2 e, respectivamente, ao receptor 3.

[00025] Módulo de controle de remetente 10 é arranjado a jusante do remetente 2 e inclui uma memória temporária de dados de remetente 12 e um processador de remetente 13. Memória temporária de dados de remetente 12, por exemplo uma fila de FIFO, está conectada ao canal de dados 7 e armazena os dados enviados pelo remetente 2. Processador de remetente 13 está conectado e adquire dados da memória temporária de remetente 12 e gera dados de controle enviados pelo canal seguro 8.

[00026] Módulo de controle de receptor 11 está arranjado a montante do receptor 3 e inclui uma memória temporária de dados de receptor 18, uma memória temporária de controle de receptor 19 e um processador de receptor 20. Memória temporária de dados de receptor 18, por exemplo uma fila de FIFO, está conectada ao canal de dados 7 e armazena os dados recebidos no canal de dados 7. Memória temporária de controle de receptor 19, por exemplo uma fila de FIFO, está conectada ao canal seguro 8 e armazena a informação de controle recebida no canal seguro 8. Memória temporária de controle de receptor 19 pode ser um componente físico ou pode ser implementada e ser inerente ao protocolo usado. Por exemplo, protocolos de

TLS (Segurança de camada de Transporte) e IPSEC (IPSecurity) têm uma memória temporária funcionando de um modo sequencial, que pode implementar a memória temporária de controle de receptor 19.

[00027] Processador de receptor 20 está conectado e adquire dados de ambas a memória temporária de dados de receptor 18 e a memória temporária de controle de receptor 19 e executa as operações necessárias para a sincronização com o módulo de controle de remetente 10, como também as operações para verificar a autenticidade dos dados transmitidos em canal de dados 7, como explicado em detalhes em seguida, com referência às Figuras 3, 4a, 4b e 4c. O processador de receptor 20 também pode estar conectado ao receptor 3 para enviar a ele mensagens de erro, de uma maneira não mostrada. Em alternativa, mensagens de erro podem ser enviadas a um servidor de administração (não mostrado) que pode tomar medidas adequadas.

[00028] O módulo de controle de remetente 10 e o módulo de controle de receptor 11 pode ser implementado como nós físicos ou como aplicativo de software que são executados por recursos já existentes das LANs 5, 6 ou da rede 4.

[00029] Figura 2 mostra uma concretização diferente, em que o módulo de controle de remetente 10 e o módulo de controle de receptor 11 estão arranjados dentro da rede de comunicação de transporte e interconexão (coluna vertebral). Por exemplo, um operador de telecomunicação pode oferecer um serviço de autenticação e integridade para fluxos de dados, inserindo os módulos de controle de remetente e receptor 10, 11 dentro de sua infra-estrutura. Por exemplo, o remetente 2 e receptor 3 podem ser telefones de VoIP conectados à Internet por redes de acesso de ADSL (Linha de Assinante Digital Assimétrica).

[00030] Em detalhes, o remetente 2 e o receptor 3 são dois nós de comunicação conectados à rede pública 4; o processador 13 de módulo de controle de remetente 10 está conectado a um servidor de identidade 26; e

processador 20 dentro do módulo de controle de receptor 11 está conectado a um servidor de registro/contabilidade 27.

[00031] O servidor de identidade 26 permite ao processador 13 do módulo de controle de remetente 10 identificar o remetente 2 que está gerando o fluxo de dados, por exemplo mapeando o endereço de IP usado para transmitir pacotes em uma rede de acesso de ADSL (Linha de Assinante Digital Assimétrica). O servidor de registro/contabilidade 27 tem o objetivo de levar em conta os vários eventos, tais como os erros de autenticação e os dados de contabilidade de processamento.

[00032] O módulo de controle de remetente 10 e o módulo de controle de receptor 11 podem ter função operacional semelhante àquelas dos mesmos elementos na Figura 1; assim, a operação dos dois sistemas será descrita em seguida para ambos os sistemas.

[00033] Na descrição seguinte, é assumido que o remetente 2 começa uma transmissão para o receptor 3 a um certo tempo inicial e envia um fluxo de dados de um tal comprimento que o remetente 2 e o receptor 3 sejam comprometidos simultaneamente na comunicação, isto é, para uma porção grande da transferência da informação do remetente para o receptor, o receptor recebe os dados enquanto o remetente ainda está transmitindo. Isto implica que o remetente 2 começa a transmissão a um tempo não conhecido ao receptor 3 e o receptor 3 começa a receber os dados transmitidos quando o remetente 2 ainda está transmitindo. Tal situação é descrita na Figura 3, em que a tempo t0 o remetente 2 começa a transmissão de um fluxo de dados incluindo bytes [a_1, a_2, ..., a_i, ..., a_M]; a tempo t1 > t0, o receptor 3 começa a receber um fluxo de dados incluindo bytes [b_1, b_2, ..., b_j, ..., b_N]; a tempo t2 > t1, o remetente 2 termina a transmissão de bytes [a_1, a_2, ..., a_i, ..., a_M] (isto é, depois de enviar byte a_M); e a tempo t3 > t2, o último byte b_N é recebido pelo receptor 3.

[00034] O presente método de autenticação é visado em verificar que

$[a_1, a_2, ..., a_i, ..., a_M]$; = $[b_1, b_2, ..., b_j, ..., b_N]$, mas para vários erros que são permitidos pelo sistema (por exemplo, perda ou deterioração de um número limitado de bytes definidos pelo sistema ser permissível). Tais erros podem ser detectados e sinalizados pelo método de autenticação. Para este fim, o fluxo de dados gerado pelo remetente 2 é transmitido inalterado pelo canal de dados 7.

[00035] Além disso, o módulo de controle de remetente 10 divide o fluxo enviado pelo remetente 2 em uma pluralidade de blocos $A_s = [a_s, ..., a_{s+L}]$, cada bloco incluindo L unidades de mensagem, por exemplo quatro unidades de mensagem, calcula um valor de autenticação para cada bloco e envia o valor de autenticação pelo canal seguro 8 para o módulo de controle de receptor 11. Aqui, o termo "unidades de mensagem" se refere, em geral, a bytes; porém, em transmissões como pacote, pode se referir a pacotes. Por causa de simplicidade, na descrição seguinte, referência será feita a byte, a menos que especificado diferentemente.

[00036] O valor de autenticação de cada bloco é calculado usando uma função *hash* H. Como conhecido, uma função *hash* é uma transformação que gera uma carreira de tamanho fixo que é "difícil de inverter" (quer dizer, dado um valor de *hash* h, é computacionalmente impossível achar uma entrada x tal que H(x) = h) e é resistente à colisão (quer dizer, é computacionalmente impossível achar duas entradas x, y, tal que H(x) = H(y)).

[00037] O módulo de controle de receptor 11 divide o fluxo recebido pelo canal de dados 7 em blocos, calcula um valor de autenticação próprio dos blocos recebidos e compara o valor de autenticação próprio com o valor de autenticação recebido por canal seguro 8 para verificar a integridade dos blocos recebidos.

[00038] A fim de permitir transmissão de fluxo de dados de comprimento desconhecido, enviado a um momento desconhecido pelo remetente para o receptor, de acordo com um aspecto da invenção, os

módulos de controle de remetente e receptor 10, 11 executam uma fase de sincronização, como abaixo discutido em detalhes em seguida, com referência ao fluxograma das Figuras 4a-4c, como também às Figuras 1, 2. Em particular, Figura 4a se refere às operações executadas pelo módulo de controle de remetente 10 e Figuras 4b e 4c se referem às operações executadas pelo módulo de controle de receptor 11.

[00039] Na Figura 4a, o módulo de controle de remetente 10 é ativado assim que se torna ciente da transmissão de um fluxo de dados [a_1, a_2, ..., a_i, ..., a_M], etapa 30; então, o fluxo de dados é remetido, inalterado, no canal de dados 7 e é duplicado simultaneamente e carregado na memória temporária de remetente 12. Na alternativa, o fluxo de dados pode ser copiado na memória temporária de remetente 12 em qualquer momento adequado. Depois da acumulação de m bytes, o processador de remetente 13 extrai uma subsequência de k bytes consecutivos P1 = [a_k, ..., a_{k+p}], chamado um padrão, com k+p < m, em que p é um número fixo independente de m (por exemplo, p = 1024), etapa 32.

[00040] O padrão P1 é selecionado preferivelmente assim para minimizar a probabilidade que o padrão P1 seja igual a outra subsequência do fluxo. Seleção do padrão também depende do tipo de tráfego; em particular, o padrão pode ser uma sequência de dados/byte cujo comprimento reduz a probabilidade de uma colisão. De acordo com uma concretização preferida, o processador de remetente 13 extrai uma sequência de bytes de tamanho relativamente grande (por exemplo, 8 pacotes de 500 bytes) dos primeiros blocos a serem verificados pelo sistema de autenticação, assim há uma probabilidade muito pequena de colisão.

[00041] O processador de remetente também pode verificar que o padrão selecionado P1 não é só formado de sequências de bytes padrão, por exemplo, de uma carreira representando uma palavra de um idioma natural ou similar.

[00042] Então, etapa 34, o processador de remetente 13 gera uma primeira mensagem de controle contendo dados de sincronização I1, uma função *hash* H e o comprimento L dos blocos. Em particular, os dados de sincronização I1 podem ser o mesmo padrão selecionado P1 ou qualquer informação que identifica univocamente o padrão P1. Por exemplo, em protocolos que associam um número às mensagens dentro de um fluxo de dados, tal como o RTP - Protocolo em Tempo Real - que provê um número de sequência, os dados de sincronização I1 podem ser o número de sequência de protocolo.

[00043] Em uma concretização, a mensagem de controle pode incluir um comando de fim, por exemplo instruindo o controle de módulo de receptor 11 para interromper o processo de autenticação depois de um dado número N de blocos ou depois de um dado tempo (por exemplo, depois de dez minutos) ou ao receber um comando ou informação específica.

[00044] A primeira mensagem de controle [I1, H, L] é então enviada sobre o canal seguro 8, etapa 38. Em particular, a primeira mensagem de controle pode ser transmitida usando quaisquer dos sistemas bem conhecidos (por exemplo protocolo de TLS - Segurança de camada de Transporte - ou IPSEC).

[00045] Além disso, o fluxo de dados gerado pelo remetente 2 é dividido em blocos $A_s = [a_s, ..., a_{s+L}]$ tendo o comprimento L enviado com a primeira mensagem de controle e cada bloco é acumulado na memória temporária de remetente 12.

[00046] Assim que um bloco $A_s = [a_s, ..., a_{s+L}]$ é acumulado na memória temporária de remetente 12, o processador de remetente 13 o carrega, etapa 40, calcula o valor de *hash* h_s usando a função *hash* H enviada com a primeira mensagem de controle, etapa 42, e envia uma mensagem de autenticação no canal seguro 8, etapa 44. Em particular, depois de enviar a primeira mensagem de controle, o processador de remetente 13 adquire o

bloco [a_{k+p+1}, ..., a_{k+p+L}], seguindo o padrão selecionado.

[00047] De acordo com uma primeira concretização, a mensagem de autenticação inclui o valor de *hash* calculado h_s. De acordo com uma concretização diferente, o processador de remetente 13 envia o valor de *hash* h_s junto com um símbolo de autenticação.

[00048] O processo de adquirir um bloco, enquanto calcular o valor de *hash* disso e enviar a mensagem de autenticação no canal seguro 8 (etapas 40-44) pode ser repetido para o fluxo inteiro, assim gerando um fluxo de controle [h_s, h_{s+L}, h_{s+2L}, ...].

[00049] As etapas descritas são então repetidas até que a transmissão do fluxo de dados original termine (saída sim da etapa 46) ou até que o processador de remetente 13 receba um pedido de re-sincronização do processador de receptor 20 (saída sim da etapa 48). No caso anterior, o processador de remetente 13 recebe uma segunda mensagem de controle semelhante à primeira mensagem de controle e incluindo novos dados de sincronização 12, referidos a um novo padrão de sincronização P2, a função *hash* H, e o comprimento L. Então, o processador de remetente 13 executa re-sincronização, identificando, na memória temporária de remetente 12, o bloco B_z seguindo padrão P2, bloco 50 e retoma o procedimento de transmissão da etapa 42, calculando o valor de *hash* do bloco B_z.

[00050] Figura 4b mostra o fluxograma das operações executadas pelo processador de receptor 20, assim que o módulo de controle de receptor 11 começa a receber uma transmissão. Em particular, o módulo de receptor 11 pode ser ativado assim que os primeiros bytes de um fluxo de dados [b_1, b_2, ..., b_N] é recebido do canal de dados 7.

[00051] Quando o módulo de receptor é ativado, o fluxo de dados recebido é acumulado na memória temporária de dados de receptor 18. Na prática, de acordo com uma concretização, o fluxo de dados recebido de canal 7 e dirigido ao receptor 3 é duplicado e o fluxo de dados duplicado é

armazenado na memória temporária de dados recebidos 18.

[00052] Assim que a mensagem de controle [I1, H, L] é recebida pela memória temporária de controle de receptor 19 de canal 8, o processador de receptor 20 a carrega, etapa 58; depois disso, o fluxo de controle [h_s, h_{s+L}, h_{s+2L}, ...], é acumulado na memória temporária de controle de receptor 19.

[00053] Depois de receber a primeira mensagem de controle, o processador de receptor 20 carrega as subsequências $B_i = [b_i, ..., b_{i+L}]$, com $1 < i < h-L$ e b_h último byte recebido, etapa 60, e compara a subsequência carregada com o padrão P1, etapa 62. Na alternativa, se os dados de sincronização recebidos I1 forem um número de sequência, como provido para o protocolo usado, o processador de receptor 20 procura um bloco recebido tendo o mesmo número de sequência.

[00054] As etapas descritas são repetidas até que o processador de receptor 20 ache uma subsequência $B_q = [b_q, ..., b_{q+L}]$ que é igual ao padrão P1 (ou cujo número de sequência de protocolo é igual aos dados de sincronização I1), saída "sim" da etapa 62. Agora, o módulo de receptor 11 está sincronizado com o módulo de remetente 10.

[00055] Depois disso, o processador de receptor 20 carrega o bloco $B_s = [b_s, ..., b_{s+L}]$ seguindo o usado para sincronização da memória temporária de dados de receptor 18, etapa 64, calcula o valor de *hash* f_s disso pela função *hash* recebida H, etapa 66, carrega o valor de *hash* recebido h_s da memória temporária de controle de receptor 19, etapa 68, e compara f_s com h_s, etapa 70. Se as mensagens de controle contiverem um símbolo de autenticação, este símbolo é verificado por meio padrão, por exemplo usando assinatura digital simétrica baseada em segredos compartilhados pelo módulo de remetente 10 e pelo módulo de receptor 11 (por exemplo, de acordo com as indicações da Assinatura Digital de XML padrão).

[00056] Se o resultado da comparação for positivo, isto é, se $f_s = h_s$, segue que $[b_s, ..., b_{s+L}] = [a_s, ..., a_{s+L}]$, e o bloco recebido B_s é autenticado.

Portanto, o receptor está assegurado que o bloco recebido foi enviado pelo remetente assumido (autenticidade das partes) e não foi modificado durante a transmissão (integridade da mensagem).

[00057] Se as verificações na etapa 70 derem resultados positivos, o procedimento anterior (etapas 64-70) continua com os blocos seguintes B_s até o fim do fluxo de dados (saída sim da etapa 72), se não (saída não da etapa 70), um procedimento de erro é implementado (etapa 74).

[00058] A situação de erro pode ser administrada de modos diferentes, levando em conta o tipo de erro. De acordo com uma primeira solução, o receptor pode apenas enviar uma mensagem de sinalização de erro ambos ao servidor de registro/contabilidade 27 (na concretização da Figura 2) e ao módulo de remetente 10, sem interromper a transmissão. De acordo com uma segunda solução, o sistema pode aceitar vários erros antes de parar a transmissão. De acordo com uma terceira solução, o sistema pode interromper imediatamente a transmissão.

[00059] A primeira e segunda soluções podem ser vantajosas quando o canal de transmissão usado é conhecido estar defeituoso. Este é o caso por exemplo de redes com bandas de transmissão muito baixas ou de redes sem fios que são estruturalmente inseguras (por exemplo, rede de IPv4 através de HF). Neste exemplo, muitos protocolos de comunicação, por exemplo todos os protocolos baseados em TCP e SIP (Protocolo de Iniciação de Sessão), provêem retransmitir automaticamente as mensagens perdidas. Por outro lado, quando retransmissão não é provida (por exemplo, de acordo com Protocolo de Datagrama de Usuário - UDP), o sistema pode incluir uma tolerância a erro. Por exemplo, se a rede de comunicação for conhecida perder uma média de 5% dos pacotes de IP enviados, o sistema pode aceitar 5% de erros de autenticação, assumindo que tais erros são devido a erros de transmissão e não a problemas de segurança. Além disso se, durante a transmissão, o módulo de receptor 11 perder sua sincronização com o módulo de transmissor

10, um procedimento de re-sincronização pode ser começado.

[00060] Figura 4c descreve o fluxograma de um possível procedimento de administração de erro, com reinício de sincronização.

[00061] Em particular, inicialmente, o processador de receptor 20 gera uma mensagem de erro para seu servidor de registro/contabilidade 27, para a concretização da Figura 2 e/ou para o módulo de remetente 10, etapa 80. Então, o tipo de erro é verificado, etapa 82, para descobrir se o erro era devido à perda de um ou mais pacotes ou a um dano nos dados recebidos. Em particular, se a transmissão prover um número de sequência, o número de sequência dos pacotes recebidos pode ser verificado; se a transmissão não prover números de sequência, os valores de *hash* dos blocos seguintes são verificados.

[00062] Se o erro for devido à perda de pacotes, saída "sim" da etapa 82, o processador de receptor 20 envia um pedido de re-sincronização, etapa 84. Em particular, o pedido de re-sincronização inclui dados de controle I2 para um novo padrão de sincronização P2, a função *hash* H e o comprimento L. Então, o processador de receptor 20 volta à etapa 64 da Figura 4b, para verificar um bloco B_s seguindo o usado para gerar os dados de controle I2.

[00063] Se o erro não for devido à perda de pacotes, por exemplo, devido a um ataque à integridade de dados, saída "não" da etapa 82, re-sincronização geralmente não é necessária, e o processador de receptor 20 pode prosseguir e verificar o bloco seguinte, etapa 86. Verificação é feita como acima descrito, calculando o valor de *hash* e o comparando com o valor de *hash* seguinte recebido no canal seguro 8. Se o resultado da verificação da etapa 86 for positivo, saída "sim" da etapa 88, então o fluxo padrão é retomado da etapa 64 da Figura 4b; se não, saída "não" da etapa 88, o processador de receptor 20 envia uma mensagem de erro para ambos seus servidores 27, para a concretização da Figura 2, e/ou para o módulo de remetente 10, etapa 90; então bloqueia os pacotes entrando no módulo de

receptor 11. O verificação dos blocos seguintes pode ser repetida, por exemplo, algumas vezes, baseado na porcentagem de erros aceitos pelo sistema; além disso, uma mensagem de erro também pode ser enviada depois da primeira detecção de um erro (antes de verificar uma mensagem seguinte na etapa 86).

[00064] Figura 5 mostra o fluxo de dados trocados em uma comunicação baseada em um protocolo de RTP, em que cada mensagem (pacote) tem um próprio número de sequência. Assim, aqui o comprimento L enviado pelo módulo de remetente 10 para o módulo de receptor 11 se refere a pacotes. A transmissão pode considerar as imagens captadas por uma câmera de vídeo de vigilância, como enviadas através de uma rede de IP para um centro de controle.

[00065] Na Figura 5, a tempo t0, o remetente 2 envia um primeiro pacote (mensagem tendo número de sequência 2), que ativa o módulo de remetente 10. O primeiro pacote assim não é autenticado. Este pacote (como também os pacotes transmitidos sucessivos) é recebido pelo módulo de receptor 11 com um atraso Δt.

[00066] Depois de ter recebido o primeiro pacote, o processador de remetente 13 ativa o procedimento de autenticação e envia uma primeira mensagem de controle C1=<I,L,H> no canal seguro 8. No exemplo, os dados de sincronização I são o número de sequência da mensagem de RTP (aqui 2), L=4, e a função *hash* é H=SHA-256. Além disso, o módulo de remetente 10 começa a carregar sua memória temporária 12 com os pacotes na ordem que são enviados e, depois de carregar quatro pacotes (primeiro bloco), calcula o valor de *hash* h1 para o primeiro bloco. Então, o processador de remetente 13 envia o valor de *hash* h1 no canal seguro 8.

[00067] Enquanto isso, o módulo de receptor 11 recebeu a primeira mensagem de controle C1 e começa a receber e carregar sua memória temporária de dados 18 com os pacotes recebidos. Assim que o módulo de

receptor 11 recebeu o número comunicado de pacotes formando um bloco (aqui, quatro), o processador de receptor 20 calcula a função *hash* disso, h1'. Então, o processador de receptor 20 compara h1 e h1'. Se eles casarem, a integridade de pacotes 3-6 é verificada positivamente.

[00068] O processo é repetido para pacotes 7-10, 11-14, ..., até o fim do fluxo.

[00069] O sistema e método como descritos têm as vantagens seguintes.

[00070] O remetente e o receptor não requerem nenhuma adaptação, desde que o fluxo de dados original não é modificado, por esse meio permitindo autenticação de dados também em aplicativos de legado.

[00071] O método pode ser aplicado a todos os protocolos de comunicação do tipo de pacote, ambos para fluxos de dados discretos e contínuos.

[00072] Desde que os módulos de controle 10, 11 estão separados ambos do remetente e do receptor, eles podem ser implementados de um modo modular e flexível, como componentes adicionais, de acordo com a política de segurança particular.

[00073] A solução descrita não introduz nenhum atraso ao fluxo de dados transmitidos, como a transmissão dos dados de autenticação é efetuada em um canal paralelo, sem interferir com a elaboração do fluxo de dados original. Este aspecto é particularmente importante para transmissão em tempo real, que não pode ser autenticada por métodos requerendo a modificação do fluxo original.

[00074] Finalmente, está claro que numerosas modificações e variantes podem ser feitas ao presente método e sistema, tudo caindo dentro da extensão da invenção, como definida nas reivindicações anexas.

[00075] Em particular, a mensagem de sincronização trocada entre o módulo de remetente 10 e o módulo de receptor 11 pode não conter a função

hash H e o comprimento L, se o sistema prover uma única função *hash* H e comprimento L (por exemplo, eles estão conectados por fios no sistema), e/ou podem conter informação adicional, tal como um identificador de módulo de remetente e o símbolo de autenticação, para provar sua identidade.

[00076] Além disso, os dados de sincronização I podem incluir dados diferentes, por exemplo porções não contíguas de um padrão, uma informação de identificação de remetente e similar.

[00077] O procedimento de erro também pode ser diferente do descrito, e inclui interrupção imediata da transmissão, pedidos de reenviar pacotes perdidos, assim por diante.

[00078] Se o fluxo de dados não contiver um número alto de dados, autenticação pode ser executada dado por dado, sem dividir o fluxo de dados em blocos.

REIVINDICAÇÕES

1. Método para autenticação fora de banda de fluxos de dados transmitidos através de uma rede de comunicação (4) incluindo um remetente (2); um receptor (3); um módulo de controle de remetente (10) e um módulo de controle de receptor (11), caracterizado pelo fato de incluir as etapas de:

transmitir um primeiro fluxo de dados através de um primeiro canal (7) conectando o remetente (2) com o receptor (3);

receber, por dito módulo de controle de remetente (10), dito primeiro fluxo de dados de dito remetente (2);

gerar dados de autenticação de dito primeiro fluxo de dados por dito módulo de controle de remetente (10);

transmitir ditos dados de autenticação do módulo de controle de remetente (10) para o módulo de controle de receptor (11) através de um segundo canal (8) conectando o módulo de controle de remetente (10) com o módulo de controle de receptor (11); e

verificar autenticidade de um segundo fluxo de dados recebidos através de dito primeiro canal (7) por dito módulo de controle de receptor (11) usando ditos dados de autenticação,

adicionalmente incluir a etapa de trocar uma mensagem de controle incluindo dados de sincronização entre o módulo de controle de remetente (10) e o módulo de controle de receptor (11) através de dito segundo canal (8);

em que a etapa de trocar uma mensagem de controle inclui uma etapa de extração, em que o módulo de remetente (10) extrai um padrão de dito primeiro fluxo de dados, ditos dados de sincronização identificando univocamente dito padrão extraído.

2. Método, de acordo com a reivindicação 1, caracterizado pelo fato de que a etapa de trocar uma mensagem de controle é executada antes de transmitir ditos dados de autenticação.

3. Método, de acordo com a reivindicação 1, caracterizado pelo fato de que a etapa de verificar a autenticidade do segundo fluxo de dados inclui uma etapa de sincronização, em que o módulo de controle de receptor (11) procura dito padrão extraído em dito segundo fluxo de dados na base de ditos dados de sincronização.

4. Método, de acordo com a reivindicação 1 ou 3, caracterizado pelo fato de que os ditos dados de sincronização incluem dito padrão extraído ou um número de sequência de pacote.

5. Método, de acordo com qualquer uma das reivindicações 1 a 4, caracterizado pelo fato de que a etapa de gerar dados de autenticação inclui a etapa de calcular, por dito módulo de controle de remetente (10), primeiros valores de *hash* de primeiras porções de dito primeiro fluxo de dados seguindo dito padrão extraído, e a etapa de transmitir dados de autenticação inclui a etapa de transmitir ditos valores de *hash* primeiro através de dito segundo canal (8),

e em que a etapa de verificar autenticidade do segundo fluxo de dados inclui carregar no módulo de controle de receptor (11) o segundo fluxo de dados, calcular segundos valores de *hash* para segundas porções do segundo fluxo de dados e comparar ditos segundos valores de *hash* com os primeiros valores de *hash* recebidos.

6. Método, de acordo com a reivindicação 5, caracterizado pelo fato de que a dita mensagem de controle também inclui uma função *hash*, e em que dito primeiro e segundo valores de *hash* são calculados por dito módulo de controle de remetente (10) e dito módulo de controle de receptor (11) usando dita função *hash*.

7. Método, de acordo com qualquer uma das reivindicações 1 a 6, caracterizado pelo fato de que a etapa de receber um primeiro fluxo de dados inclui dividir dito primeiro fluxo de dados em primeiros blocos de um comprimento fixo;

a etapa de gerar dados de autenticação inclui calcular valores de *hash* de ditos primeiros blocos;

a etapa de verificar autenticidade do fluxo recebido de dados inclui dividir o segundo fluxo de dados em segundos blocos de um comprimento fixo, calcular valores de *hash* de ditos segundos blocos, e comparar ditos dados de autenticação com ditos valores de *hash* de ditos segundos blocos.

8. Método, de acordo com a reivindicação 7, caracterizado pelo fato de que a etapa de receber um primeiro fluxo de dados adicionalmente inclui acumular ditos primeiros blocos em uma memória temporária de remetente (12);

a etapa de gerar dados de autenticação inclui carregar um bloco de ditos primeiros blocos de dita memória temporária de remetente (12);

a etapa de verificar autenticidade do fluxo recebido de dados inclui acumular ditos segundos blocos em uma memória temporária de dados de receptor (18), acumular ditos dados de autenticação em uma memória temporária de controle de receptor (19) e carregar um bloco de ditos segundos blocos de dita memória temporária de dados de receptor (18).

9. Método, de acordo com qualquer uma das reivindicações 1 a 8, caracterizado pelo fato de que a dita mensagem de controle adicionalmente inclui uma informação de comprimento de ditos padrões.

10. Método, de acordo com qualquer uma das reivindicações 1 a 9, caracterizado pelo fato de que se durante dita etapa de verificar autenticidade do segundo fluxo de dados, o módulo de controle de receptor (11) detectar um erro entre ditos dados de autenticação e o segundo fluxo de dados, o módulo de controle de receptor (11) envia um pedido de re-sincronização para dito módulo de controle de remetente (10).

11. Método, de acordo com a reivindicação 10, caracterizado

pelo fato de que o dito pedido de re-sincronização inclui segundos dados de sincronização identificando univocamente um segundo padrão extraído do segundo fluxo de dados.

12. Método, de acordo com a reivindicação 11, caracterizado pelo fato de que os segundos dados de sincronização incluem um valor de *hash* de dito segundo padrão extraído do segundo fluxo de dados.

13. Sistema para autenticação fora de banda de fluxos de dados transmitidos através de uma rede de comunicação (4), caracterizado pelo fato de incluir:

um módulo de controle de remetente (10) configurado para receber um primeiro fluxo de dados de um remetente (2) e gerar dados de autenticação;

um módulo de controle de receptor (11) configurado para verificar ditos dados de autenticação;

um primeiro canal (7) conectando dito remetente (2) a dito receptor (3) configurado para transmitir dito primeiro fluxo de dados de dito remetente para dito receptor;

um segundo canal (8) conectando dito módulo de remetente (10) e dito módulo de receptor (11) configurado para transmitir ditos dados de autenticação;

em que dito módulo de controle de remetente (10) e dito módulo de controle de receptor (11) incluem unidades de sincronização respectivas (12, 13; 18-20) trocando mensagem de controle incluindo dados de sincronização; e

em que a dita unidade de sincronização de dito módulo de controle de remetente (10) inclui uma memória temporária de remetente (12) configurada para carregar primeiras porções do primeiro fluxo de dados e um processador de remetente (13) configurado para extrair um padrão (P1) de ditas primeiras porções e transmitir dita mensagem de controle para dito

módulo de controle de receptor (11) através de dito segundo canal (8), ditos dados de sincronização (I1) identificando univocamente dito padrão extraído (P1).

14. Sistema, de acordo com a reivindicação 13, caracterizado pelo fato de que a dita unidade de sincronização de dito módulo de controle de receptor (11) inclui uma memória temporária de dados de receptor (18) configurada para carregar segundas porções de um segundo fluxo de dados recebidos através de dito primeiro canal (7) e um processador de receptor (20) configurado para receber dita mensagem de controle de dito segundo canal (8) e para receber ditas segundas porções de dita memória temporária de dados de receptor (18), em que dito processador de receptor (20) é configurado para procurar dito padrão extraído de ditas segundas porções na base de ditos dados de sincronização.

15. Sistema, de acordo com a reivindicação 13 ou 14, caracterizado pelo fato de que o dito módulo de controle de remetente (10) faz parte de uma primeira rede local (5) incluindo dito remetente (2) e dito módulo de controle de receptor (11) faz parte de uma segunda rede local (6) incluindo dito receptor (3).

16. Sistema, de acordo com qualquer uma das reivindicações 13 a 15, caracterizado pelo fato de que o dito módulo de controle de remetente (10) e dito módulo de controle de receptor (11) fazem parte de dita rede de comunicação (4).

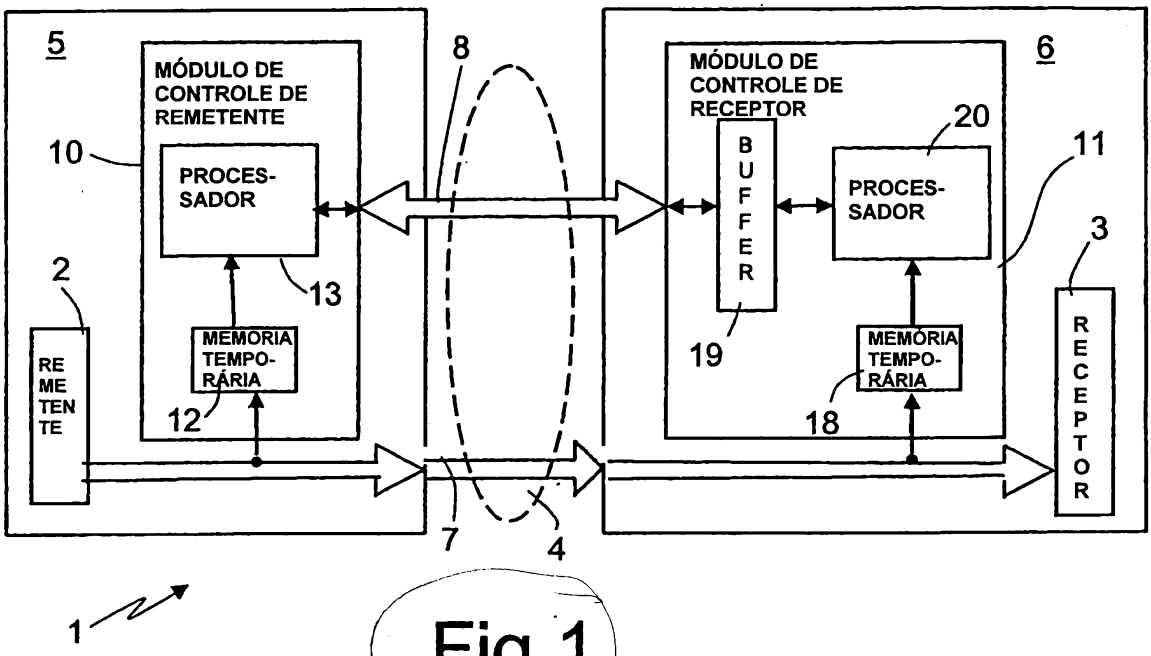
5
MÓDULO DE CONTROLE DE REMETENTE
10
PROCES-SADOR
13
MEMÓRIA TEMPO-RÁRIA
12
2
REMETENTE
8
7
4
MÓDULO DE CONTROLE DE RECEPTOR
6
BUFFER
19
20
PROCES-SADOR
11
3
MEMÓRIA TEMPO-RÁRIA
18
RECEPTOR
1

Fig.1

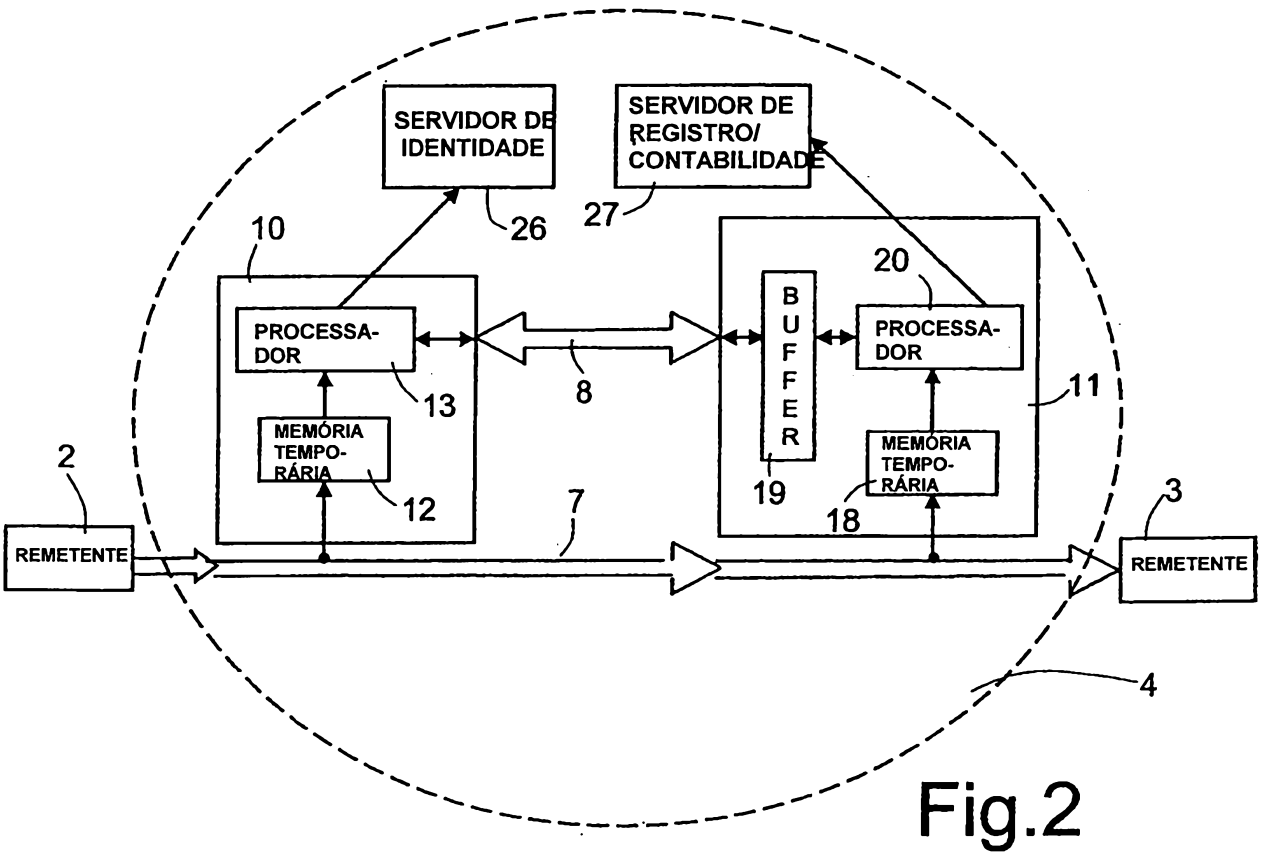
SERVIDOR DE IDENTIDADE
26
SERVIDOR DE REGISTRO/ CONTABILIDADE
27
10
PROCESSA-DOR
13
MEMÓRIA TEMPO-RÁRIA
12
8
BUFFER
19
20
PROCESSA-DOR
11
MEMÓRIA TEMPO-RÁRIA
18
2
REMETENTE
7
3
REMETENTE
4

Fig.2

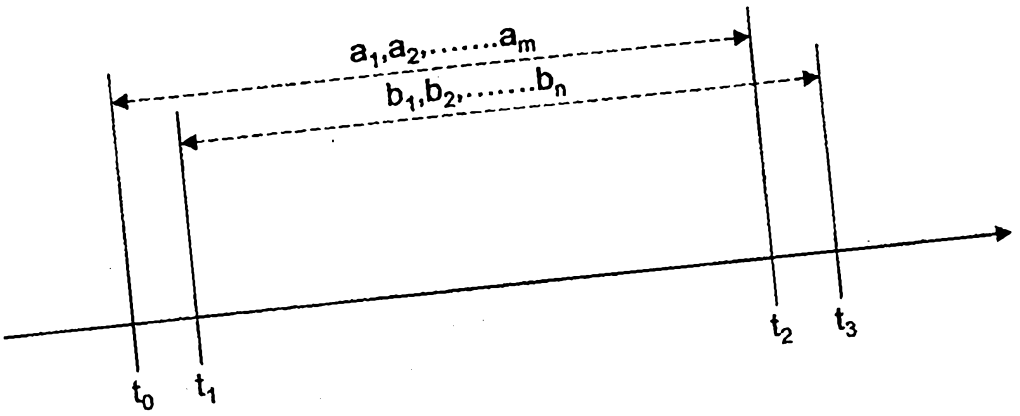
a1,a2,.....am
b1,b2,.....bn
t0
t1
t2
t3

Fig.3

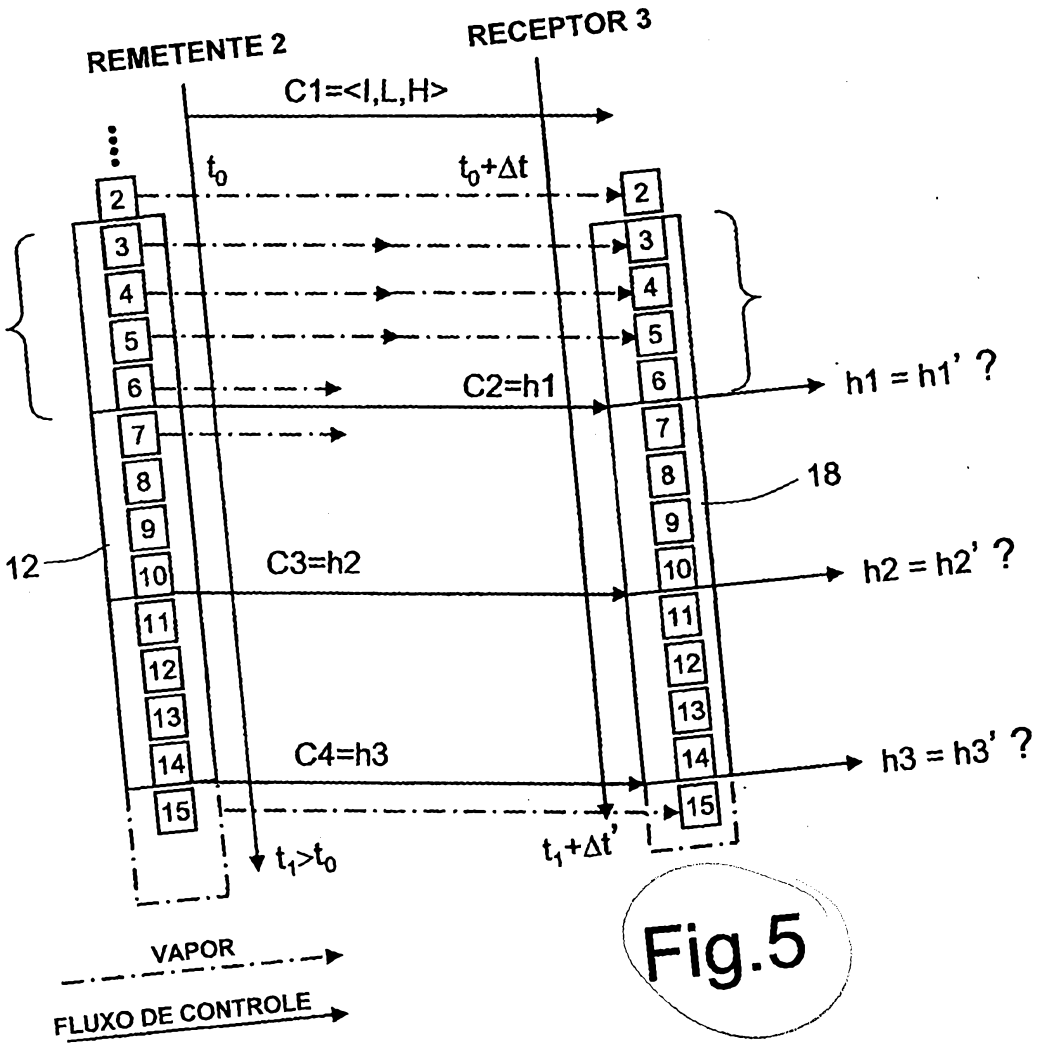
REMETENTE 2
RECEPTOR 3
C1=<I,L,H>
t0
t0+Δt
C2=h1
h1 = h1' ?
18
12
C3=h2
h2 = h2' ?
C4=h3
h3 = h3' ?
t1>t0
t1+Δt'
VAPOR
FLUXO DE CONTROLE

Fig.5

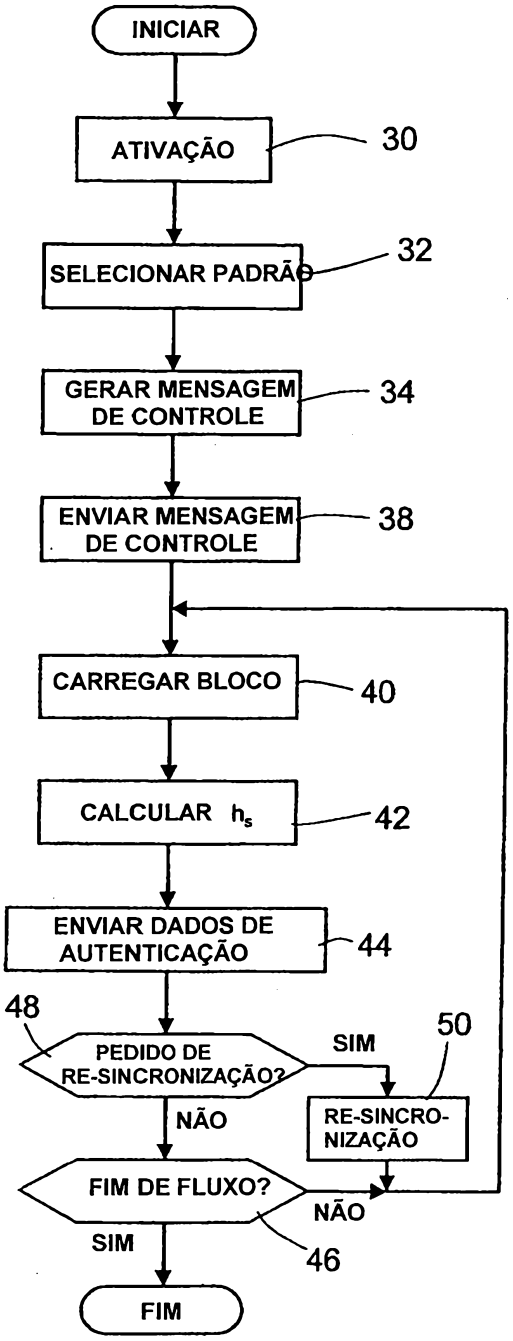
INICIAR
ATIVAÇÃO
30
SELECIONAR PADRÃO
32
GERAR MENSAGEM DE CONTROLE
34
ENVIAR MENSAGEM DE CONTROLE
38
CARREGAR BLOCO
40
CALCULAR h_s
42
ENVIAR DADOS DE AUTENTICAÇÃO
44
48
PEDIDO DE RE-SINCRONIZAÇÃO?
SIM
50
RE-SINCRO-NIZAÇÃO
NÃO
FIM DE FLUXO?
NÃO
SIM
46
FIM

Fig.4a

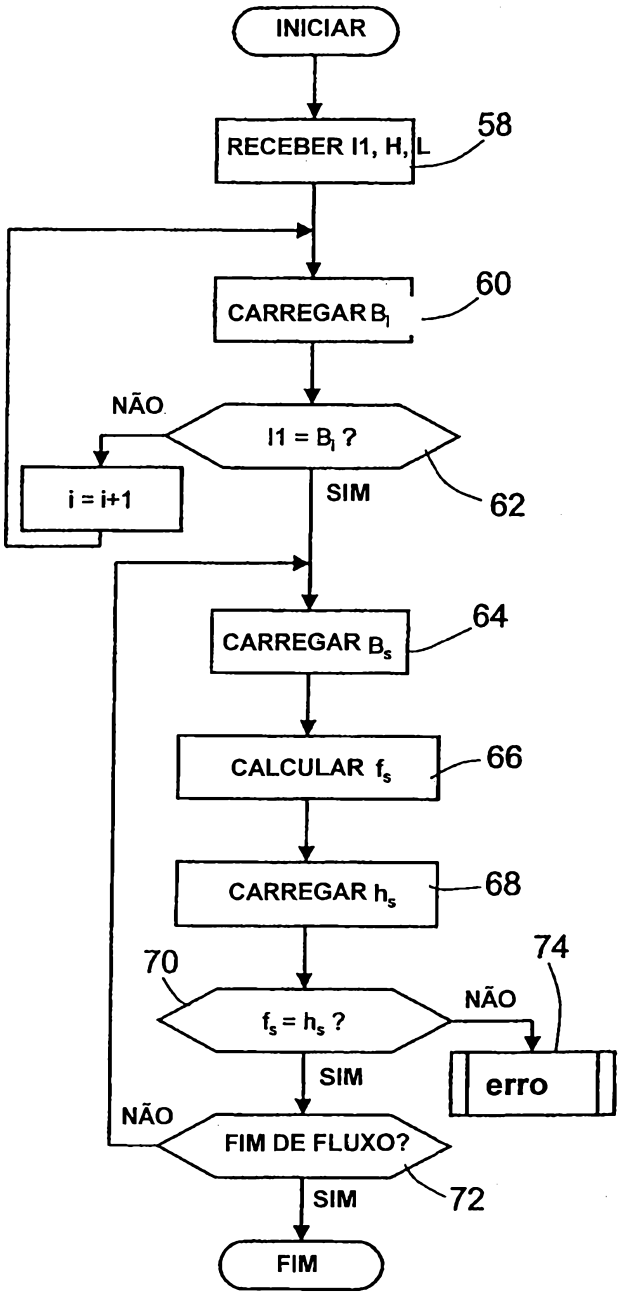
INICIAR
RECEBER I1, H, L
58
CARREGAR B_I
60
NÃO
I1 = B_I ?
62
i = i+1
SIM
CARREGAR B_s
64
CALCULAR f_s
66
CARREGAR h_s
68
70
74
f_s = h_s ?
NÃO
erro
SIM
NÃO
FIM DE FLUXO?
72
SIM
FIM

Fig.4b

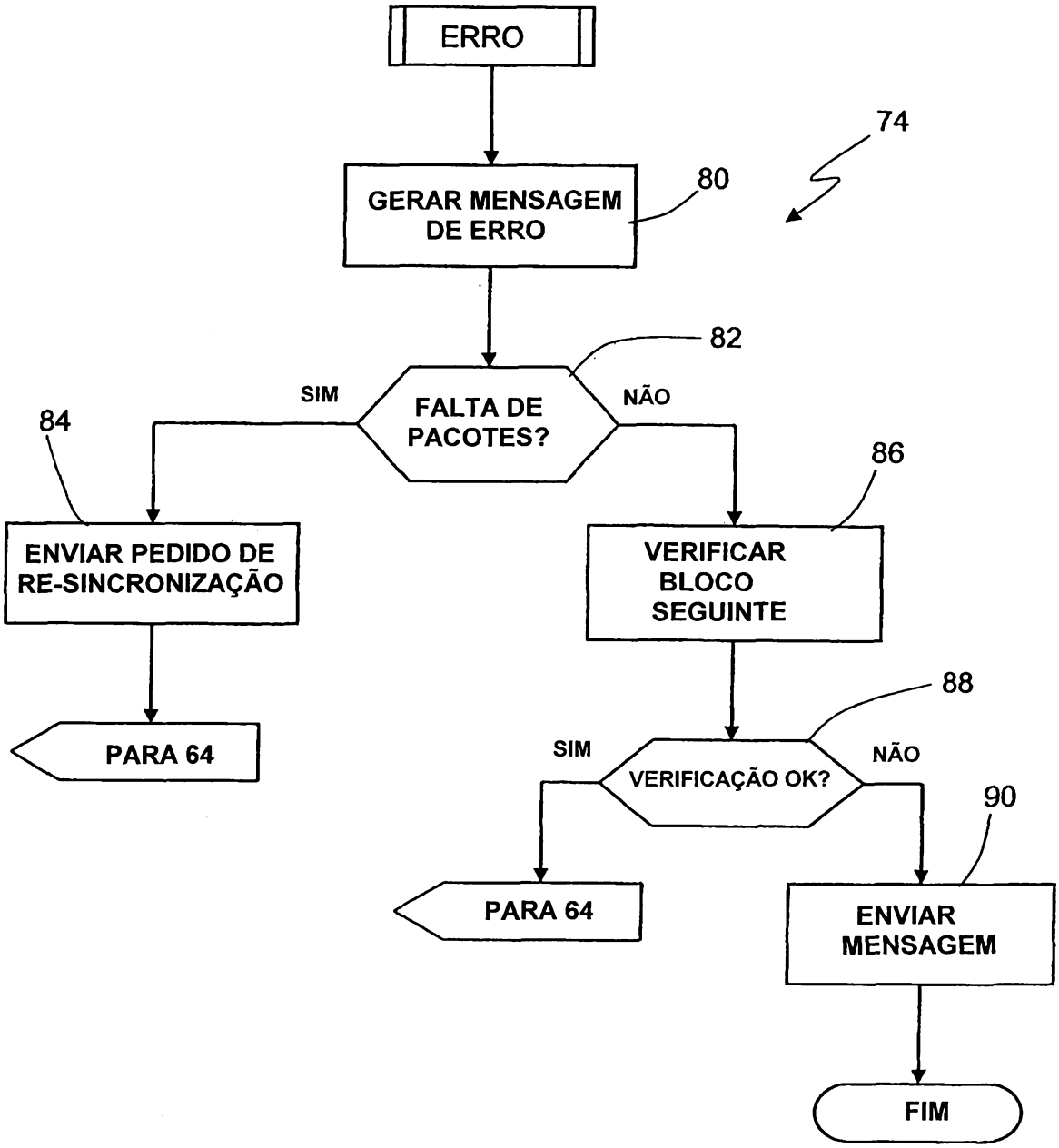
ERRO
74
GERAR MENSAGEM DE ERRO
80
82
SIM
FALTA DE PACOTES?
NÃO
84
86
ENVIAR PEDIDO DE RE-SINCRONIZAÇÃO
VERIFICAR BLOCO SEGUINTE
88
PARA 64
SIM
VERIFICAÇÃO OK?
NÃO
90
PARA 64
ENVIAR MENSAGEM
FIM

Fig.4c