



República Federativa do Brasil
Ministério da Indústria, Comércio Exterior
e Serviços
Instituto Nacional da Propriedade Industrial

(11) PI 0501325-9 B1

(22) Data do Depósito: 18/04/2005

(45) Data de Concessão: 04/09/2018



(54) Título: RENDERIZAR CONTEÚDO DIGITAL EM UM SISTEMA DE PROTEÇÃO DE CONTEÚDO DE ACORDO COM UMA PLURALIDADE DE LICENÇAS DIGITAIS ENCADEADAS

(51) Int.Cl.: H04L 9/32

(30) Prioridade Unionista: 22/04/2004 US 10/831.281

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

(72) Inventor(es): BRIAN P. EVANS; CLIFFORD P. STROM; MICHAEL JAY PARKS

(85) Data do Início da Fase Nacional: 18/04/2005

"RENDERIZAR CONTEÚDO DIGITAL EM UM SISTEMA DE PROTEÇÃO DE CONTEÚDO DE ACORDO COM UMA PLURALIDADE DE LICENÇAS DIGITAIS ENCADEADAS"

CAMPO TÉCNICO

5 A presente invenção relaciona-se com uma arquitetura e método para permitir que o conteúdo digital seja sintetizado em um sistema de proteção de conteúdo de acordo com uma pluralidade de licenças digitais encadeadas, onde cada licença no encadeamento deve ser satisfeita para realmente
10 renderizar o conteúdo. Mais particularmente, a presente invenção relaciona-se com tal arquitetura e método por meio dos quais o conteúdo e a licença no encadeamento incluem referências apropriadas um para o outro.

FUNDAMENTO DA INVENÇÃO

15 Como é conhecido e referindo-se agora à Fig. 1, um sistema de proteção e reforço de conteúdo é altamente desejável em conexão com o conteúdo digital 12 tal como áudio digital, vídeo digital, texto digital, dados digitais, multimídia digital, etc., onde o tal conteúdo digital 12 é para
20 ser distribuído para os usuários. Quando sendo recebido pelo usuário, tal usuário sintetiza ou 'executa' o conteúdo digital com a ajuda de um dispositivo de sintetização apropriado tal como um executor de mídia em um computador pessoal 14, um dispositivo de execução portátil ou coisa parecida.

25 Tipicamente, um proprietário de conteúdo distribuindo tal conteúdo digital 12 deseja restringir o que o usuário pode fazer com tal conteúdo digital distribuído 12. Por exemplo, o proprietário do conteúdo pode desejar restringir

o usuário de copiar e redistribuir tal conteúdo 12 para um segundo usuário, ou pode desejar permitir que o conteúdo digital distribuído 12 seja executado somente um número limitado de vezes, somente por um certo tempo total, somente em um certo tipo de máquina, somente em um certo tipo de executor de mídia, somente por um certo tipo de usuário, etc.

Entretanto, após a distribuição ter ocorrido, tal proprietário do conteúdo possui muito pouco, se houver, controle sobre o conteúdo digital 12. Então, um sistema de proteção digital contra cópia 10 permite a sintetização ou execução controlada de formas arbitrárias do conteúdo digital 12, onde tal controle é flexível e pode ser definido pelo proprietário do conteúdo de tal conteúdo digital. Tipicamente, o conteúdo 12 é distribuído para o usuário na forma de um pacote 13 por meio de qualquer canal de distribuição apropriado. O pacote de conteúdo digital 13 como distribuído pode incluir o conteúdo digital 12 criptografado com uma chave de criptografia / decriptografia simétrica (KD), (isto é, (KD(CONTEÚDO))), bem como outras informações identificando o conteúdo, como adquirir uma licença para tal conteúdo, etc.

O sistema de proteção contra cópia baseado em confiança 10 permite que um proprietário do conteúdo digital 12 especifique regras que devem ser satisfeitas antes que o tal conteúdo digital 12 seja permitido de ser sintetizado. Tais regras podem incluir os requerimentos mencionados anteriormente e / ou outros e podem ser incorporadas dentro de uma licença digital 16 que o usuário / dispositivo de computação

do usuário 14 (aqui dentro posteriormente, tais termos podem ser alternados a menos que circunstâncias requeiram de outro modo) deve obter a partir do proprietário do conteúdo ou de um agente do mesmo, ou tais regras podem já estar ligadas com o conteúdo 12. Tal licença 16 e / ou regras podem por exemplo incluir a chave de deciptografia (KD) para deciptografar o conteúdo digital 12, talvez criptografado de acordo com outra chave que pode ser deciptografada pelo dispositivo de computação do usuário ou por outro dispositivo de execução.

O proprietário de conteúdo para uma parte de conteúdo digital 12 preferiria não distribuir o conteúdo 12 para o usuário a não ser que tal proprietário pudesse confiar em que o usuário irá submeter-se às regras especificadas pelo tal proprietário de conteúdo na licença 16 ou em qualquer outro lugar. Então, de preferência, o dispositivo de computação do usuário 14 ou outro dispositivo de execução é proporcionado com um componente ou mecanismo acreditado 18 que não irá renderizar o conteúdo digital 12 exceto de acordo com tais regras.

O componente acreditado 18 tipicamente possui um avaliador 20 que inspeciona as regras e determina baseado nas regras inspecionadas se o usuário requisitante possui o direito de renderizar o conteúdo digital requisitado 12 no modo procurado, dentre outras coisas. Como deve ser entendido, o avaliador 20 é acreditado no sistema de proteção contra cópia 10 para realizar as vontades do proprietário do conteúdo digital 12 de acordo com as regras e o usuário não

deve estar apto a facilmente alterar tal componente acreditado 18 e / ou o avaliador 20 para qualquer propósito, nefário ou de outro modo.

Como deve ser entendido, as regras para renderizar o conteúdo 12 podem especificar se o usuário possui direitos para então renderizar baseado em qualquer um dentre vários fatores, incluindo quem é o usuário, onde o usuário está localizado, qual tipo de dispositivo de computação 14 ou dispositivo de execução o usuário está utilizando, qual aplicação de sintetização está chamando o sistema de proteção contra cópia 10, a data, a hora, etc. Em adição, as regras podem limitar a sintetização à um número predeterminado de execuções, ou tempo predeterminado de execução, por exemplo.

As regras podem ser especificadas de acordo com qualquer linguagem e sintaxe apropriadas. Por exemplo, a linguagem pode simplesmente especificar atributos e valores que devem ser satisfeitos (a DATA deve ser posterior à X, por exemplo), ou pode requerer a performance de funções de acordo com um script especificado (SE DATA maior do que X, ENTÃO FAÇA . . . , por exemplo).

Quando o avaliador 20 determinar que as regras na licença 16 são satisfeitas, o conteúdo digital 12 pode então ser sintetizado. Em particular, para renderizar o conteúdo 12, a chave de deciptografia (KD) é obtida a partir de uma fonte predefinida tal como a licença mencionada anteriormente 16 e é aplicada junto ao (KD(CONTEÚDO)) a partir do pacote de conteúdo 13 para resultar no conteúdo real 12 e o conteúdo real 12 é então realmente sintetizado.

Observe que o componente acreditado 18 pode às vezes ser requerido para manter a informação de estado relevante para a sintetização de uma parte de conteúdo particular 12 e / ou para o uso de uma licença particular 16. Por exemplo, pode ser o caso em que uma licença particular 16 possui um requerimento de contagem de execução e por consequência o componente acreditado 18 deve se lembrar quantas vezes a licença 16 foi empregada para renderizar o conteúdo correspondente 12 ou quantas vezes a mais a licença 16 pode ser empregada para renderizar o conteúdo correspondente 12. Por consequência, o componente acreditado 18 também pode incluir pelo menos um armazenamento seguro persistente 22 dentro do qual tal informação de estado é de forma persistente mantida de um modo seguro. Assim, o componente confiável 18 armazena tal informação de estado no tal armazenamento seguro 22 de um modo persistente de modo que a tal informação de estado seja mantida mesmo através de sessões de uso no dispositivo de computação 14. Tal armazenamento seguro 22 está provavelmente localizado no dispositivo de computação 14 do componente acreditado 18, apesar do tal armazenamento seguro 22 poder estar alternativamente localizado em qualquer outro lugar.

Em um sistema de proteção contra cópia 10, o conteúdo 12 é empacotado para uso por um usuário por criptografar tal conteúdo 12 e associar um conjunto de regras com o conteúdo 12, por meio do que o conteúdo 12 pode ser sintetizado somente de acordo com as regras. Então, devido ao fato do conteúdo 12 somente poder ser sintetizado de acordo com

as regras, o conteúdo 12 pode ser livremente distribuído. Tipicamente, o conteúdo 12 é criptografado de acordo com uma chave simétrica tal como a chave mencionada anteriormente (KD) para resultar em (KD(contéudo)) e, portanto, 5 (KD(contéudo)) também é decriptografada de acordo com (KD) para resultar no conteúdo 12. Tal (KD) é por sua vez incluída dentro da licença 16 correspondendo ao conteúdo 12. Frequentemente, tal (KD) é criptografada de acordo com uma chave pública tal como a chave pública do dispositivo de computação 14 (PU-C) no qual o conteúdo 12 é para ser sintetizado, resultando em (PU-C(KD)). Contudo, observe que outras chaves públicas podem ser empregadas, tal como por exemplo uma chave pública de um usuário, uma chave pública de um grupo do qual o usuário é um membro, etc.

15 Somente uma única licença 16 foi requerida até agora para renderizar o conteúdo correspondente 12. Tal única licença 16 está tipicamente ligada com um único usuário, com uma única máquina, com um único grupo de usuários ou coisa parecida, como representado pela criptografia de chave pública (KD) dentro de tal licença. De forma correspondente, 20 somente um possuidor da chave privada correspondendo à tal chave pública pode acessar (KD), supondo obviamente que um componente acreditado 18 permite isto. Entretanto, é para ser apreciado que existem casos em que pode ser vantajoso 25 requerer várias licenças 16 para renderizar tal conteúdo 12.

Para um exemplo, uma das várias licenças 16 pode especificar uma primeira de duas partes das regras para renderizar o conteúdo 12 e outra das várias licenças 16 pode

especificar uma segunda das duas partes das regras para renderizar o conteúdo 12. Para outro exemplo, por requerer várias licenças 16 para renderizar o conteúdo 12, uma licença 16 poderia ser ligada com uma ou mais licenças 16, assim formando uma seqüência ou 'encadeamento' de licenças ligadas ou 'encadeadas' 16 levando à uma licença raiz 16. Em tal situação e como deve ser apreciado, as regras em cada licença encadeada 16 devem ser satisfeitas para permitir que o conteúdo correspondente 12 seja sintetizado.

10 Contudo, observe que não existe arquitetura ou método para definir como ligar uma licença encadeada 16 com outra licença encadeada 16, como percorrer de uma licença encadeada 16 até a próxima ou como empregar um encadeamento de licenças 16 para renderizar o conteúdo 12. Por consequência, existe a necessidade por uma arquitetura e método que
15 efetuem a ligação de uma licença encadeada 16 com outra licença encadeada 16, percorrendo a partir de uma licença encadeada 16 até a próxima e empregando um encadeamento de licenças 16 para renderizar o conteúdo 12, dentre outras coisas.
20

SUMÁRIO DA INVENÇÃO

As necessidades mencionadas anteriormente são satisfeitas pelo menos em parte pela presente invenção na qual um método é proporcionado para renderizar conteúdo digital
25 criptografado em um dispositivo de computação de acordo com um encadeamento de licenças. No método, uma requisição para renderizar o conteúdo é recebida e o encadeamento de licenças correspondendo ao conteúdo é localizado. O encadeamento

inclui uma licença folha ligada com o conteúdo em uma extremidade do encadeamento, uma licença raiz na outra extremidade do encadeamento e quaisquer licenças intermediárias entre as mesmas. Cada uma dentre a licença folha e quaisquer licenças intermediárias no encadeamento é ligada com a licença adjacente no encadeamento em direção à licença raiz e a licença raiz é ligada com um proprietário de uma chave privada (PR-U).

Para cada licença no encadeamento, a licença é verificada e é confirmado que a licença permite que o conteúdo seja sintetizado. Uma chave de decryptografia para decryptografar o conteúdo é obtida a partir da licença folha baseado na aplicação da (PR-U) junto à licença raiz e na decryptografia de chaves de encadeamento dentro do encadeamento, a chave obtida é aplicada junto ao conteúdo criptografado para decryptografar o mesmo e o conteúdo decryptografado é sintetizado.

BREVE DESCRIÇÃO DOS DESENHOS

O sumário precedente, bem como a descrição detalhada seguinte das modalidades da presente invenção, serão melhor entendidos quando lidos em conjunto com os desenhos anexos. Para o propósito de ilustrar a invenção, são apresentadas nos desenhos modalidades que são presentemente preferidas. Entretanto, como deve ser entendido, a invenção não está limitada às disposições e instrumentalidades precisas apresentadas. Nos desenhos:

A Fig. 1 é um diagrama de blocos apresentando uma arquitetura de reforço de um exemplo de um sistema baseado

em confiança, incluindo um dispositivo de computação com um componente acreditado para receber uma licença digital correspondendo ao conteúdo digital e permitir a sintetização do conteúdo somente de acordo com a licença;

5 A Fig. 2 é um diagrama de blocos representando um sistema de computador de propósito geral no qual os aspectos da presente invenção e / ou partes da mesma podem ser incorporados.

10 A Fig. 3 é um diagrama de blocos apresentando a relação entre o conteúdo digital e a licença digital da Fig. 1 quando a licença digital é desencadeada;

15 A Fig. 4 é um diagrama de blocos apresentando a relação entre o conteúdo digital da Fig. 1 e uma pluralidade de licenças digitais encadeadas de acordo com uma modalidade da presente invenção; e

20 A Fig. 5 é um fluxograma apresentando etapas chave executadas pelo componente acreditado da Fig. 1 ao se percorrer o encadeamento da Fig. 4 no decorrer da determinação quanto a se permitir que o conteúdo seja sintetizado de acordo com uma modalidade da presente invenção.

DESCRIÇÃO DETALHADA DA INVENÇÃO

AMBIENTE DE COMPUTADOR

25 A Fig. 1 e a seguinte discussão são pretendidas para proporcionar uma breve descrição geral de um ambiente de computação adequado no qual a presente invenção e / ou partes da mesma podem ser implementadas. Apesar de não ser requerido, a invenção é descrita no contexto geral de instruções executáveis por computador, tais como módulos de

programa, sendo executados por um computador, tal como uma estação de trabalho cliente ou um servidor. Geralmente, os módulo de programa incluem rotinas, programas, objetos, componentes, estruturas de dados e coisa parecida que executam 5 tarefas particulares ou implementam tipos de dados abstratos articulares. Além disso, deve ser apreciado que a invenção e / ou partes da mesma podem ser praticadas com outras configurações de sistema de computador, incluindo dispositivos portáteis, sistemas com múltiplos processadores, componentes 10 eletrônicos baseados em microprocessador ou programáveis do consumidor, PCs da rede, minicomputadores, computadores de grande porte e coisa parecida. A invenção também pode ser praticada em ambientes de computação distribuída onde as tarefas são executadas por dispositivos de processamento remoto 15 que estão ligados através de uma rede de comunicações. Em um ambiente de computação distribuída, os módulos de programa podem estar localizados nos dispositivos de armazenamento em memória tanto local como remota.

Como apresentado na Fig. 2, um sistema de computação 20 de propósito geral ilustrativo inclui um computador pessoal convencional 120 ou coisa parecida, incluindo uma unidade de processamento 121, uma memória do sistema 122 e um barramento do sistema 123 que acopla vários componentes do sistema incluindo a memória do sistema com a unidade de processamento 121. O barramento do sistema 123 pode ser qual- 25 quer um dentre vários tipos de estruturas de barramentos incluindo um barramento de memória ou controlador de memória, um barramento periférico e um barramento local utilizando

qualquer uma dentre uma variedade de arquiteturas de barramento. A memória do sistema inclui a memória somente leitura (ROM) 124 e a memória de acesso randômico (RAM) 125. Um sistema básico de entrada / saída 126 (BIOS), contendo as rotinas básicas que ajudam a transferir informações entre os elementos dentro do computador pessoal 120, tal como durante a inicialização, é armazenado na ROM 124.

O computador pessoal 120 pode adicionalmente incluir um controlador de disco rígido 127 para leitura e gravação junto a um disco rígido (não apresentado), um controlador de disco magnético 128 para leitura e gravação junto a um disco magnético removível 129 e um controlador de disco ótico 130 para leitura e gravação junto a um disco ótico removível 131 tal como um CD-ROM ou outro meio ótico. O controlador de disco rígido 127, o controlador de disco magnético 128 e o controlador de disco ótico 130 estão conectados com o barramento do sistema 123 por uma interface do controlador de disco rígido 132, por uma interface do controlador de disco magnético 133 e por uma interface do controlador ótico 134, respectivamente. Os controlador e seus meios legíveis por computador associados proporcionam o armazenamento não volátil de instruções legíveis por computador, estruturas de dados, módulos de programa e de outros dados para o computador pessoal 120.

Apesar da modalidade ilustrativa descrita aqui dentro empregar um disco rígido, um disco magnético removível 129 e um disco ótico removível 131, deve ser apreciado que outros tipos de meios legíveis por computador que possam

armazenar dados que sejam acessíveis por um computador também podem ser utilizados no ambiente operacional ilustrativo. Tais outros tipos de meios incluem um cassete magnético, um cartão de memória flash, um disco de vídeo digital, um
5 cartucho de Bernoulli, uma memória de acesso randômico (RAM) e uma memória somente leitura (ROM) e coisa parecida.

Uma série de módulos de programa podem ser armazenados no disco rígido, no disco magnético 129, no disco ótico 131, na ROM 124 ou na RAM 125, incluindo um sistema operacional 135, um ou mais programas de aplicação 136, outros
10 módulos de programa 137 e dados de programa 138. Um usuário pode entrar com comandos e informação no computador pessoal 120 através de dispositivos de entrada tais como um teclado 140 e o dispositivo de apontamento 142. Outros dispositivos
15 de entrada (não apresentados) podem incluir um joystick, plataforma de jogo, antena de satélite, digitalizador ou coisa parecida. Estes e outros dispositivos de entrada estão freqüentemente conectados com a unidade de processamento 121 através de uma interface de portal serial 146 que está aco-
20 plada com o barramento do sistema, mas podem ser conectados por outras interfaces, tal como uma porta paralela, porta de jogo ou barramento serial universal (USB). Um monitor 147 ou outro tipo de dispositivo de exibição também está conectado com o barramento do sistema 123 via uma interface, tal como
25 um adaptador de vídeo 148. Em adição ao monitor 147, um computador pessoal tipicamente inclui outros dispositivos periféricos de saída (não apresentados), tais como alto-falantes e impressoras. O sistema ilustrativo da Fig. 2 também inclui

um adaptador hospedeiro 155, um barramento da Interface de Sistema de Computador Pequeno (SCSI) 156 e um dispositivo de armazenamento externo 162 conectado com o barramento SCSI 156.

5 O computador pessoal 120 pode operar em um ambiente em rede utilizando conexões lógicas com um ou mais computadores remotos, tal como um computador remoto 149. O computador remoto 149 pode ser outro computador pessoal, um servidor, um roteador, um PC da rede, um dispositivo par ou outro
10 tro nó comum da rede e tipicamente inclui vários ou todos os elementos descritos acima relativos ao computador pessoal 120, apesar de somente um dispositivo de armazenamento em memória 150 ter sido ilustrado na Fig. 2. As conexões lógicas descritas na Fig. 2 incluem uma rede de área local (LAN)
15 151 e uma rede de área ampla (WAN) 152. Tais ambientes em rede são comuns em escritórios, redes de computador de grandes empresas, em intranets e na Internet.

Quando utilizado em um ambiente em rede LAN, o computador pessoal 120 está conectado com a LAN 151 através
20 de uma interface ou adaptador da rede 153. Quando utilizando em um ambiente em rede WAN, o computador pessoal 120 tipicamente inclui um modem 154 ou outro dispositivo para estabelecer comunicações através da rede de área ampla 152, tal como a Internet. O modem 154 o qual pode ser interno ou ex-
25 terno, está conectado com o barramento do sistema 123 via a interface de portal serial 146. Em um ambiente em rede, os módulos de programa descritos em relação ao computador pessoal 120, ou partes dos mesmos, podem ser armazenados no

dispositivo de armazenamento em memória remota. Será apreciado que as conexões de rede apresentadas são ilustrativas e que outro dispositivo para estabelecer uma ligação de comunicação entre os computadores pode ser utilizado.

5 LICENÇAS ENCADEADAS 16 E USO DAS MESMAS

Na presente invenção, o conteúdo 12 pode ser acessado em um dispositivo de computação 14 de acordo com um sistema de proteção de conteúdo 10 que pode processar uma sequência ou 'encadeamento' das licenças ligadas ou 'encadeadas' 16.

Voltando-se agora para a Fig. 3, é visto que na situação onde uma licença única, não encadeada 16 é ligada com um único usuário, com uma única máquina, com um único grupo de usuários ou coisa parecida, uma chave pública da mesma (PU-U) criptografia (KD) dentro de tal licença 16 para resultar em (PU-U(KD)). Assim, somente um possuidor da chave privada (PR-U) correspondendo à tal (PU-U) pode acessar (KD) a partir de tal (PU-U(KD)), supondo sem dúvidas que um componente acreditado 18 permite isto. Em prática, tal licença 20 não encadeada 16 é empregada na seguinte maneira. De forma preliminar, um usuário seleciona um pacote de conteúdo 13 com uma parte de conteúdo 12 correspondendo à licença 16, onde o conteúdo 12 no pacote 13 já está criptografado de acordo com uma chave de conteúdo (KD) para resultar em 25 (KD(contéudo)). O pacote 13 também possui uma ID de conteúdo ou 'KID' que identifica o conteúdo 12. Em pelo menos algumas instâncias, tal KID pode representar um valor a partir do qual a chave de conteúdo (KD) para decriptografar o conteúdo

12 pode ser derivada.

De qualquer maneira, tal KID é incluída com a licença não encadeada 16, como é apresentado na Fig. 3 e a licença 16 é presumivelmente armazenada dentro de um armazenamento de licenças ou coisa parecida (não apresentado) e é indexada dentro do armazenamento de licenças de acordo com a KID no mesmo. Assim, quando da seleção do pacote de conteúdo 13 com a KID na mesma, o componente acreditado 18 pode empregar tal KID para localizar e selecionar a licença 16 no armazenamento de licenças.

Tal licença 16 presumivelmente inclui com a mesma uma assinatura digital baseada nos conteúdos da licença 16 e a chave privada de um emissor da licença (PR-L) e assim tal assinatura pode ser verificada pela aplicação de uma chave pública correspondente do emissor (PU-L) de um modo que deve ser conhecido ou aparente para o público relevante, onde (PU-L) está presumivelmente disponível para o componente acreditado 18. Por consequência, quando da seleção da licença 16, o componente acreditado 18 realmente verifica tal assinatura, e no caso em que a verificação é positiva, o uso da licença 16 para renderizar o conteúdo correspondente 12 continua.

Em particular e como foi relatado acima, a licença 16 também inclui regras ou política que devem ser avaliadas pelo avaliador de licença 20 e satisfeitas antes de permitir que o conteúdo 12 seja sintetizado. Assim, o avaliador de licença 20 realmente avalia as regras na licença 16 para determinar se as regras permitem que o conteúdo 12 seja sinte-

tizado na maneira procurada. Supondo que as regras realmente permitem que o conteúdo 12 seja sintetizado, então, o componente acreditado 18 localiza a (PU-U(KD)) como relatada na licença 16 e aplica a (PR-U) ao mesmo para revelar (KD), onde (PR-U) está presumivelmente disponível para o componente acreditado. Como agora deve ser evidente, o componente acreditado 18 pode então aplicar (KD) junto ao (KD(contéudo)) a partir do pacote de conteúdo 13 para revelar o conteúdo 12 e então envia o conteúdo revelado 12 para uma aplicação apropriada (não apresentada) para sintetização pela mesma.

Novamente, a licença única, não encadeada 16 da Fig. 3 está diretamente ligada com o usuário, máquina, grupo do usuário, etc. por meio da utilização da chave pública (PU-U) do mesmo para criptografar a chave de conteúdo (KD) e por consequência nenhuma outra licença 16 é necessária para renderizar o conteúdo 12 correspondendo à tal licença única 16.

Entretanto, e voltando-se agora para a Fig. 4, em uma modalidade da presente invenção, uma licença encadeada 16 está ligada com outra licença 16 para formar um encadeamento 24 de tais licenças 16, por meio do qual cada licença 16 no encadeamento 24 deve estar presente e estar verificada e as regras em cada licença 'encadeada' 16 devem ser satisfeitas para permitir que o conteúdo correspondente 12 seja sintetizado. Em tal modalidade, e como apresentado, tal encadeamento 24 inclui:

- uma licença 'folha' 16I com a KID do conteúdo correspondente 12 (KID1, aqui);

- uma licença 'raiz' 16r que está ligada com o u-

suário, máquina, grupo do usuário ou coisa parecida por meio da utilização da chave pública (PU-U) da mesma para criptografar um valor que deve ser decriptografado; e

5 zero, uma ou mais licenças intermediárias 16i (uma sendo apresentada na Fig. 4) interconectando a licença folha 16I e a licença raiz 16r para formar o encadeamento 24.

Como visto e como com uma licença não encadeada 16, cada uma da licença folha 16I, licença folha 16r e qualquer licença intermediária 16i inclui uma KID (KID1, KID2, 10 KID3, etc.), um conjunto de regras e uma assinatura digital.

Como também visto, a licença folha 16I e quaisquer licenças intermediárias 16i no encadeamento 24 são distinguidas de uma licença não encadeada 16 pelo fato de que cada tal licença 16I, 16i adicionalmente inclui uma $KID(x + 1)$ de 15 ligação ascendente apontando para a próxima licença 16 no encadeamento 24 em direção à licença raiz 16r, seja ela uma licença intermediária 16i ou a licença raiz 16r. A licença folha 16I e quaisquer licenças intermediárias 16i no encadeamento 24 também são distinguidas de uma licença não encadeada 16 pelo fato de que uma chave de criptografia criptografando a outra chave criptográfica não é $(PU-U(KD))$. Ao invés disso, no caso da licença folha 16I, (KD) é criptografada de acordo com uma chave de ligação simétrica (KLx) . Como apresentado, tal (KLx) está disponível a partir da próxima li- 20 cença 16 no encadeamento 24 em direção à licença raiz 16r. De forma similar, no caso de qualquer licença intermediária 16i, uma chave de ligação simétrica $(KL(x-1))$ para ser empregada e conexão com a próxima licença 16 no encadeamento 25

24 em direção à licença folha 16I é criptografada de acordo com outra chave de ligação simétrica (KL_x), onde outra (KL_x) está novamente disponível a partir da próxima licença 16 no encadeamento 24 em direção à licença raiz 16r.

5 Como adicionalmente visto, a licença raiz 16r no encadeamento 24 não inclui qualquer $KID(X+1)$ de ligação ascendente, especialmente porque a licença raiz 16r está na extremidade do encadeamento 24. De forma significativa, na licença raiz 16r, a uma chave de criptografia criptografando
10 a outra chave de criptografia não é $(PU-U(KD))$, mas ao invés disso é $(PU-U)$ criptografando a chave de ligação simétrica $(KL(x-1))$ para ser empregada em conexão com a próxima licença 16 no encadeamento 24 em direção à licença folha 16I (isto é, $(PU-U(KL(x-1)))$).

15 Como agora deve ser apreciado, com o encadeamento 24 das licenças 16 como até aqui descrito, cada licença 16, seja ela uma licença folha 16I ou uma licença intermediária 16i, é encadeada para a próxima licença 16 em direção à licença raiz 16r por incluir na mesma a $KID(x+1)$ de ligação
20 ascendente para tal próxima licença 16. Em adição, a licença raiz 16r está ligada com um usuário, máquina, grupo de usuários, etc. pela $(PU-U(KL(x-1)))$ na mesma. De forma correspondente, cada licença 16, seja ela uma licença raiz 16r ou uma licença intermediária 16i, é encadeada para a próxima
25 licença 16 em direção à licença folha 16I por incluir na mesma a chave de ligação simétrica $(KL(x-1))$ para tal próxima licença 16. Finalmente, é para ser lembrado que a licença folha 16I possui a chave de conteúdo (KD) para decriptogra-

far o conteúdo protegido 12.

Agora, como pode ser apreciado, o encadeamento 24 das licenças 16 como relatado em conexão com a Fig. 4 é empregado por seguir a KID1 em uma parte de conteúdo 12 até a
 5 licença folha 16I possuindo tal KID1 e então por seguir a KID(X+1) de ligação ascendente (KID1, KID2, KID3) em cada licença 16 do encadeamento 14 até uma licença raiz 16r sem tal KID(X+1) de ligação ascendente ser encontrada. Após isto, (PR-U) é aplicado junto a (PU-U(KL(x-1))) a partir da
 10 licença raiz 16r para revelar a chave de ligação (KL(x-1)) para a próxima licença 16 em direção à licença folha 16I e o encadeamento 24 é percorrido na direção oposta, durante o que, em cada etapa, uma (KLx(KL(x-1))) é decriptografada para revelar (KL(x-1)) se a licença 16 for uma licença inter-
 15 mediária 16i ou senão uma (KL1(KD)) é decriptografada para revelar (KD) se a licença 16 for a licença folha 16I.

De forma significativa, cada licença 16 em um encadeamento particular 24 é avaliado de forma independente com respeito a se a assinatura digital da mesma confere e se
 20 as regras da mesma permitem a sintetização do conteúdo 12 no modo procurado. Então, para um encadeamento de licença 24 permitir a sintetização de conteúdo, cada licença 16 deve permitir tal sintetização. Os dados de estado em um armazenamento de dados seguro 22 tal como contagens de ação são
 25 mantidos por licença 16 e são manipulados independentemente para cada licença 16 em um encadeamento 24. Por exemplo, se uma licença raiz 16r e uma licença folha 16I formarem um encadeamento completo 24 para o conteúdo particular 12 e cada

uma possuir uma contagem de execução, tanto a contagem de execução para a licença raiz 16r como a contagem de execução para a licença folha 16I são de forma apropriada ajustadas quando o encadeamento 24 é empregado para renderizar o conteúdo 12. Se qualquer contagem de execução impedisse que o conteúdo 12 fosse sintetizado, o conteúdo 12 realmente não é sintetizado, mesmo se a outra contagem pudesse realmente permitir que o conteúdo 12 fosse sintetizado.

Voltando-se agora para a Fig. 5, um método para empregar um encadeamento 24 de licenças tal como estas apresentadas na Fig. 4 é revelado em maiores detalhes. De forma preliminar, é suposto que cada licença 16 no encadeamento 24 da Fig. 4 esteja presente em um dispositivo de computação 14 possuindo o conteúdo 12 que é para ser sintetizado baseado em tal encadeamento 24, incluindo uma licença folha 16I com uma KID2 de ligação ascendente e uma (KL1(KD)); qualquer licença intermediária 16i com uma KID(x+1) de ligação ascendente (KID3, KID4, etc.) e uma (KLx(KL(x-1))); e uma licença raiz 16r com uma (PU-U(KL(x-1))).

Quando da recepção de uma requisição para que o conteúdo 12 seja sintetizado (etapa 501), então, um componente acreditado 18 no dispositivo de computação determina uma KID1 associada com tal conteúdo 12 (etapa 503), encontra uma licença 16 em um armazenamento de licenças no dispositivo de computação possuindo tal KID1 (etapa 505) e verifica tal licença encontrada 16 e garante que as regras da licença encontrada 16 permitem a sintetização do conteúdo 12 no modo procurado (etapa 507). Observe que o componente confiável 18

não sabe ainda se a licença encontrada 16 é uma licença folha 16I ou uma licença não encadeada 16. Assim, supondo que a licença encontrada 16 verifica e permite a sintetização do conteúdo 12, o componente acreditado 18 então determina se a
5 licença encontrada 16 possui uma KID de ligação ascendente (KID2, aqui) (etapa 59). Como agora deve ser evidente, se a licença encontrada 16 não possuir uma KID2 de ligação ascendente, tal licença encontrada 16 é uma licença não encadeada 16 e não uma licença folha 16I e o processamento da licença
10 não encadeada 16 continua como apresentado na Fig. 5 através das etapas 511, 513 e então 523.

Entretanto, e também como agora deve ser evidente, se a licença encontrada 16 realmente possuir uma KID2 de ligação ascendente, tal licença encontrada 16 é uma licença
15 folha 16I e o processamento assim continua na etapa 505, onde o componente acreditado encontra uma licença 16 (uma licença intermediária 16i ou uma licença raiz 16r) e um armazenamento de licenças no dispositivo de computação possuindo tal KID2 (de ligação ascendente) e verifica tal licença en-
20 contrada 16 e garante que as regras da licença encontrada 16 permitem a sintetização do conteúdo 12 no modo procurado, como na etapa 507. Observe aqui que o componente acreditado 18 ainda não sabe se a licença encontrada 16 é uma licença intermediária 16i ou uma licença raiz 16r. Assim, supondo
25 que a licença encontrada 16 verifica e permite a sintetização do conteúdo 12, o componente acreditado 18 novamente determina se a licença encontrada 16 possui uma KID de ligação ascendente (KID3, aqui), como na etapa 509. Como deve ser

evidente aqui, se a licença encontrada 16 não possuir uma KID3 de ligação ascendente, tal licença encontrada 16 é uma licença raiz 16r e o processamento assim continua até a etapa 511 (ver abaixo) onde o encadeamento 24 das licenças 16 é percorrido para trás em direção à licença folha 16I.

Entretanto e como deve ser evidente aqui, se a licença encontrada 16 realmente possuir uma KID3 de ligação ascendente, tal licença encontrada 16 é uma licença intermediária 16i e o processamento assim continua novamente na etapa 505. Assim, uma ou mais licenças intermediárias 16i são processadas até uma licença raiz 16r ser encontrada pela ausência de uma $KID(x+1)$ de ligação ascendente na mesma na etapa 509.

Quando realmente determinando que uma licença encontrada 16 é uma licença raiz 16r, o componente acreditado 18 continua o processamento por obter a partir da licença raiz encontrada 16r a $(PU-U(KL(x-1)))$ que está incluída com a mesma (etapa 511) e aplicar $(PR-U)$ junto a tal $(PU-U(KL(x-1)))$ para revelar $(KL(x-1))$ (etapa 513). Observe que o componente acreditado 128 é assim suposto de estar na posse de tal $(PR-U)$. Observe também que, como foi relatado acima, a $(KL(x-1))$ revelada é para ser empregada com respeito à próxima licença 16 em direção à licença folha 16I. Adicionalmente, observe que para a $(KL(x-1))$ revelada ser empregada com respeito à próxima licença 16 em direção à licença folha 16I, o componente acreditado 18 deve seguir as licenças 16 no encadeamento 24 em sua ordem nas mesmas como foram encontradas quando percorrendo em direção à licença raiz 16r,

tanto por meio de manter uma lista apropriada (não apresentada) como de outro modo.

Assim, para cada licença intermediária encontrada 16i no encadeamento 24 e em ordem a partir da licença raiz 16r até a licença folha 16I, o componente acreditado 18 obtém a partir de tal licença intermediária 16i a $(KLx(KL(x-1)))$ que está incluída com a mesma (etapa 515) e aplica (KLx) junto ao $(KLx(KL(x-1)))$ para revelar $(KL(x-1))$ (etapa 517). Como agora deve ser apreciado, para qualquer licença intermediária particular 16i, a (KLx) a ser aplicada junto ao $(KLx(KL(x-1)))$ da mesma é encontrada a partir da próxima licença 16 em direção à licença raiz 16r, seja ela uma tal licença raiz 16r ou outra licença intermediária 16i.

Finalmente, após o componente acreditado 18 obter $(KL1)$ a partir da última licença intermediária 16i em direção à licença folha 16I como na etapa 517, tal componente acreditado 18 então obtém a partir da licença folha 16I a $(KL1(KD))$ que está incluída com a mesma (etapa 519), aplica $(KL1)$ junto a tal $(KL1(KD))$ para revelar (KD) (etapa 521) e aplica (KD) junto a $(KD(\text{conteúdo}))$ a partir do pacote de conteúdo 13 para revelar o conteúdo 12 (etapa 523).

Obviamente, se o encadeamento 24 não incluir a licença intermediária 16i, como é determinado na etapa 514, a licença raiz 16r inclui $(PU-U(KL1))$ e o processo continua a partir da etapa 513 diretamente para a etapa 519.

Como agora deve ser evidente a partir da Fig. 5, o processamento das licenças 16 pelo componente acreditado 18 não requer que cada licença encontrada 16 seja realmente ro-

tulada como uma licença não encadeada 16, como uma licença
 folha 16I, como uma licença intermediária 16i, como uma li-
 cença raiz 16r ou coisa parecida. Ao invés disso, é típico
 na presente invenção que: (1) uma licença 16 possuindo uma
 5 KID de uma parte de conteúdo 12 e nenhuma KID(x+1) de liga-
 ção ascendente é uma licença não encadeada 16; (2) uma li-
 cença 16 possuindo uma KID de uma parte de conteúdo 12 e uma
 KID(x+1) de ligação ascendente é uma licença folha 16I; (3)
 uma licença 16 possuindo uma KIDx referenciada pela KID(x+1)
 10 de ligação ascendente de outra licença 16 e possuindo uma
 KID(x+1) de ligação ascendente é por si própria uma licença
 intermediária 16i; e (4) uma licença 16 possuindo uma KIDx
 referenciada pela KID(x+1) de ligação ascendente de outra
 licença 16 e não possuindo uma KID(x+1) de ligação ascenden-
 15 te é por si própria uma licença raiz 16r. Contudo, observe
 que podem existir casos onde, por exemplo, uma licença 16
 pode ser tanto uma licença intermediária 16i com respeito à
 alguma parte de conteúdo 12 como uma licença folha 16I com
 respeito à outra parte de conteúdo 12. Entretanto, é presen-
 20 temente acreditado que tal caso é desaconselhável por ser
 tanto desnecessário como indevidamente confuso durante a o-
 peração.

Como até aqui relatado na presente revelação, as
 licenças 16 podem ser ligadas uma com as outras de um modo
 25 linear como um encadeamento 24. Entretanto, é para ser apre-
 ciado que as licenças 16 também podem ser ligadas uma com as
 outras de um modo hierárquico como uma árvore possuindo uma
 pluralidade de tais encadeamentos 24. Por exemplo, No tal

caso de árvore pode ser que a KID de uma licença raiz particular 16r seja a KID de ligação ascendente de cada uma dentre uma pluralidade de licenças intermediárias 16i em um primeiro sub-nível, que para cada uma da pluralidade de licenças intermediárias 16i no primeiro sub-nível, a KID das mesmas seja a KID de ligação ascendente de cada uma dentre uma pluralidade de licenças intermediárias 16i em um segundo sub-nível e que para cada uma da pluralidade de licenças intermediárias 16i no segundo sub-nível, a KID das mesmas seja a KID de ligação ascendente para cada uma dentre uma pluralidade de licenças folha 16I em um terceiro sub-nível. Além disso, adicionalmente, pode ser o caso em que existem licenças folha adicionais 16I em qualquer um do primeiro e segundo sub-níveis com KIDs de ligação ascendente se referindo à várias licenças intermediárias 16i no primeiro sub-nível e na licença raiz 16r, respectivamente. De forma notável, com respeito à qualquer licença folha 16I em tal árvore hierárquica de licenças 16, um encadeamento 24 de licenças 16 ainda pode ser encontrado entre tal licença folha 16I e a licença raiz 16r de tal árvore e por consequência, o processo relatado em conexão com a Fig. 5 ainda pode ser empregado.

Além disso, é para ser apreciado que com uma árvore organizada de forma hierárquica de licenças 16, as regras relatadas na licença raiz 16r de tal árvore devem ser satisfeitas e tal licença raiz 16r deve ser verificada de modo a renderizar qualquer parte de conteúdo 12 ligado com a árvore. De forma similar, para uma sub-ramificação de uma árvore como representada por uma licença intermediária 16I, as re-

gras relatadas em tal licença intermediária 16i de tal ramificação devem ser satisfeitas e tal licença intermediária 16i deve ser verificada de modo a renderizar qualquer parte de conteúdo 12 ligada com a ramificação. Como um resultado e
5 como pode ser apreciado, uma única licença 16 na árvore pode controlar a sintetização em conexão com várias licenças folha 16I que se ramificam a partir de tal licença única 16 dentro da árvore. Por exemplo, a revogação, expiração e coisa parecida de tal licença única 16 na árvore impediria
10 qualquer sintetização em conexão com tais licenças folha 16I que se ramificam a partir de tal licença única 16 dentro da árvore.

Em um cenário imaginado, várias licenças folha 16I são ligadas tanto diretamente como indiretamente com uma licença raiz 16r que expira mensalmente. assim para permitir à
15 um usuário visualizar o conteúdo 12 ligado com tais licenças folha 16, tal como para o conteúdo ilustrativo 12 organizado em uma biblioteca, um serviço de assinatura iria emitir novamente a licença raiz 16r cada mês para os usuários que assinam o mesmo. Como deve ser evidente, emitir novamente a
20 licença raiz 16r é muito mais simples e mais rápido do que emitir novamente cada licença folha 16I.

Observe que pode existir um caso onde, por exemplo, uma árvore inclui uma parte cíclica ou de repetição na
25 mesma. Enquanto tecnicamente praticável, tal como pode exemplificar em um caso onde a licença 16 poderia incluir várias KIDS de ligação ascendente que são acessadas condicionalmente, é presentemente acreditado que tal caso não é aconselhável por

ser tanto desnecessário como de forma indevida confuso durante a construção e a operação.

Como pode ser apreciado em conexão com a presente invenção, seria útil proporcionar dados de licença correntes durante a aquisição de uma licença folha 16I ou de uma licença intermediária 16i a cerca de outras licenças 16 que podem estar no encadeamento 24 em direção e incluindo a licença raiz 16r que são requeridos para renderizar o conteúdo correspondente 12. De modo semelhante, o conteúdo 12, o próprio conteúdo 12 pode possuir dados de licença correntes similares. Tais dados de licença podem compreender uma lista de KIDS / KIDS de ligação ascendente. Baseado em tais dados de licença, um sistema de proteção de conteúdo 10 e / ou o componente acreditado 18 do mesmo pode emitir requisições de licença por licenças requeridas 16 do encadeamento, a medida que necessário. Tais dados de licença também podem ser úteis durante a aquisição de uma licença raiz, talvez para descrever uma árvore de licenças intermediárias e folha disponíveis 16i, 16I, dentre outras coisas.

20 CONCLUSÃO

A presente invenção pode ser praticada com respeito à qualquer conteúdo apropriado 12 e as licenças 16 organizadas em qualquer modo ligado particular. Como agora deve ser apreciado, com a presente invenção como relatado aqui dentro, a sintetização do conteúdo 12 pode ser controlada de acordo com mais do que meramente uma única licença correspondente 16 e em particular pode ser controlada baseado em uma pluralidade de licenças 16, cada uma das quais deve ser

verificada e ser satisfeita.

A programação necessária para efetuar os processos em conexão com a presente invenção é relativamente direta e deve ser aparente para o público de programação relevante. Por consequência, tal programação não está ligada aqui. Então, qualquer programação particular pode ser empregada para efetuar a presente invenção sem se afastar do espírito e escopo da mesma.

Na descrição precedente, pode ser visto que a presente invenção compreende uma arquitetura ou método novo e útil que permite ligar uma licença encadeada 16 com outra licença encadeada 16, percorrer a partir de uma licença encadeada 16 até a próxima e empregar um encadeamento 24 de licenças encadeadas 16 para renderizar o conteúdo 12, dentre outras coisas. Deve ser apreciado que alterações poderiam ser feitas nas modalidades descritas acima sem se afastar dos conceitos inventivos das mesmas. Então, portanto, em geral deve ser apreciado que esta invenção não está limitada às modalidades particulares reveladas, mas é pretendida para cobrir modificações dentro do espírito e escopo da presente invenção como definido pelas reivindicações anexas.

REIVINDICAÇÕES

1. Método para renderizar conteúdo digital criptografado (12) em um dispositivo de computação de acordo com uma cadeia (24) de licenças, cada licença contendo
5 regras que devem ser satisfeitas antes de renderizar o conteúdo digital, uma assinatura digital e um identificador, KID, em que a cadeia inclui uma licença folha (16L) ligada por meio do KID da licença folha ao conteúdo em uma extremidade da cadeia, uma licença raiz (16R) na outra
10 extremidade da cadeia e pelo menos uma licença intermediária (16I) entre elas, a licença folha e cada uma das pelo menos uma licença intermediária na cadeia estão vinculadas à licença anexa na cadeia em direção à licença raiz por meio de um KID de uplink, e a licença raiz está
15 vinculada a um proprietário de uma chave privada, PR-U, por meio de uma chave pública, PU-U, o método compreendendo as etapas de:

receber (501) uma requisição para renderizar o conteúdo;

20 localizar a cadeia de licenças correspondente ao conteúdo, primeiramente ao descobrir (505) a licença folha no dispositivo de computação, a licença folha estando ligada por meio de seu KID ao conteúdo, determinando que a licença folha encontrada possui um KID de uplink e,
25 posteriormente, seguir o KID de uplink ($x + 1$) em cada licença da cadeia até que uma licença raiz sem esse KID de uplink ($x + 1$) seja encontrada;

para cada licença encontrada na cadeia, verifi-

car (507) a assinatura digital da licença e confirmar que as regras da licença estão satisfeitas;

obter a chave criptográfica para descriptografar o conteúdo da licença folha, **caracterizado pelo fato de**
 5 **que** ainda compreende:

obter (511), a partir da licença raiz, uma chave de link, $KL(x-1)$ da licença adjacente $(x-1)$ -enésima para a licença folha criptografada de acordo com a chave pública, $PU-U$, para resultar em $PU-U(KL(x-1))$;

10 aplicar (513) a chave privada, $PR-U$, correspondente a $PU-U$ para $PU-U(KL(x-1))$ para revelar $KL(x-1)$;

para cada licença encontrada na cadeia entre a licença folha e a licença raiz, e na ordem a partir
 15 da licença raiz para a licença folha:

obter (515), a partir da (x) -enésima licença, uma chave de link da $(x-1)$ -enésima licença adjacente para a licença folha $KL(x-1)$ criptografada de acordo com uma chave de link da (x) -enésima licença, KLx , resultar em $KLx(KL(x-1))$; e
 20

aplicar (517) KLx a $KLx(KL(x-1))$ para revelar $KL(x-1)$, KLx sendo revelado em conexão com a $(x+1)$ -enésima licença na cadeia;

até que $KL1$ seja revelado;

25 obter (519), a partir da licença folha, uma chave de conteúdo, KD , para descriptografar o conteúdo criptografado de acordo com $KL1$ para resultar em $KL1(KD)$; e

aplicar (521) KL1 a KL1 (KD) para revelar KD;

aplicar (523) KD ao conteúdo criptografado para descriptografar o mesmo; e

5 renderizar o conteúdo descriptografado.

2. Método, de acordo com a reivindicação 1, **caracterizado pelo fato de que** as etapas de localizar a cadeia de licenças correspondente ao conteúdo, verificar as assinaturas digitais das licenças e confirmar que as regras dos certificados estão preenchidas compreendem:

determinar (503) um ID de conteúdo associado ao conteúdo;

encontrar (505) a licença folha no dispositivo de computação tendo tal ID de conteúdo como KID, sendo a
15 licença folha uma primeira licença;

verificar a assinatura digital da primeira licença encontrada e confirmar que as regras da primeira licença encontrada foram satisfeitas; e

repetidamente:

20 encontrar uma (x)-enésima licença no dispositivo de computação tendo o KID de uplink da (x-1)-enésima licença encontrada;

verificar a assinatura digital da (x)-enésima licença encontrada e confirmar que as regras da
25 (x)-enésima licença encontrada estão satisfeitas; e

determinar se a (x)-enésima licença encontrada tem um KID de uplink;

até que seja determinado que a (x)-enésima

licença encontrada não possui um KID de uplink e, portanto, é a licença raiz.

3. Método, de acordo com a reivindicação 2, **caracterizado pelo fato de que** a cadeia de licenças inclui
5 exatamente uma licença intermediária.

4. Método, de acordo com a reivindicação 2, **caracterizado pelo fato de que** a cadeia de licenças inclui exatamente duas licenças intermediárias.

5. Método, de acordo com a reivindicação 2, **caracterizado pelo fato de que** a cadeia de licenças inclui
10 exatamente três licenças intermediárias..



Fig. 1

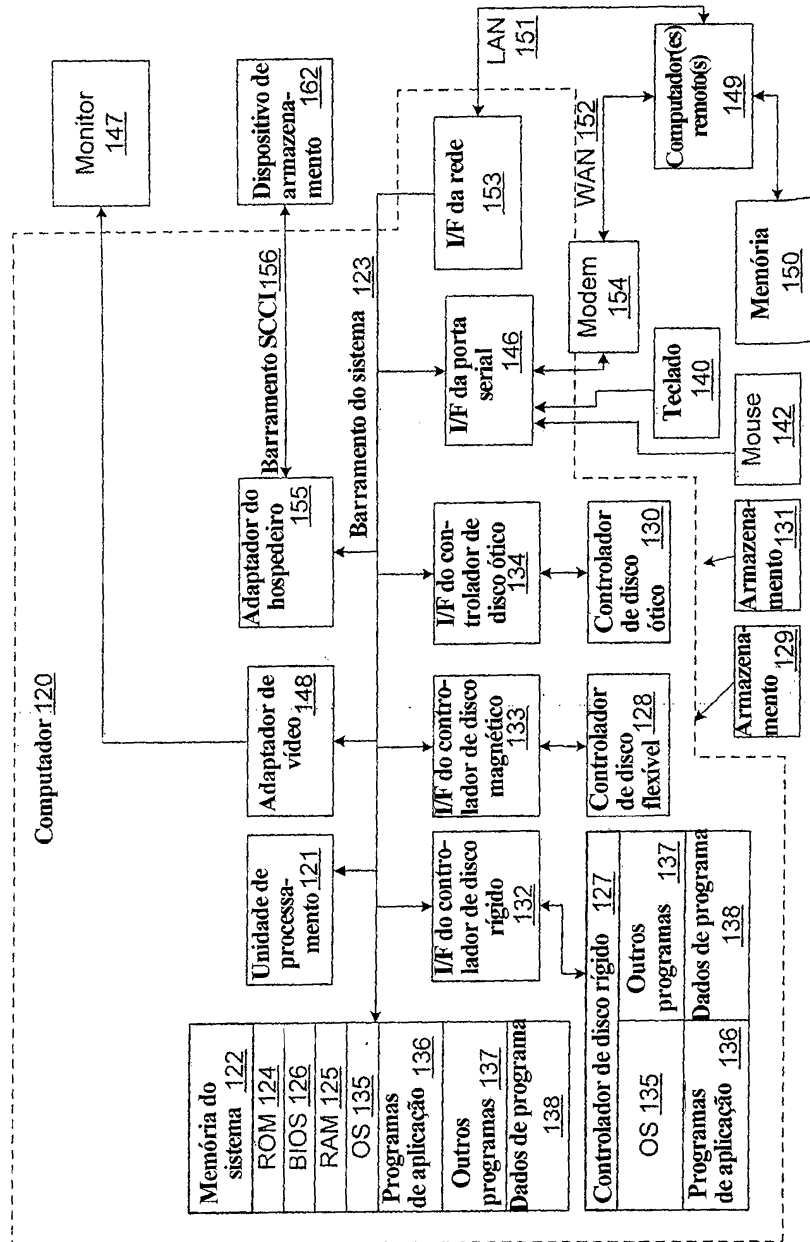
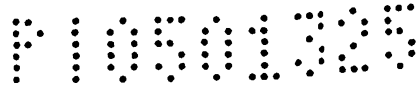


Fig. 2

45

46

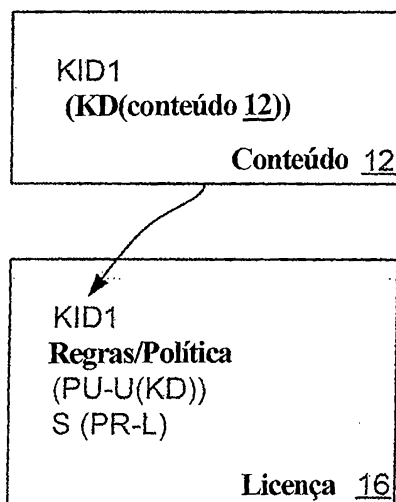


FIG. 3

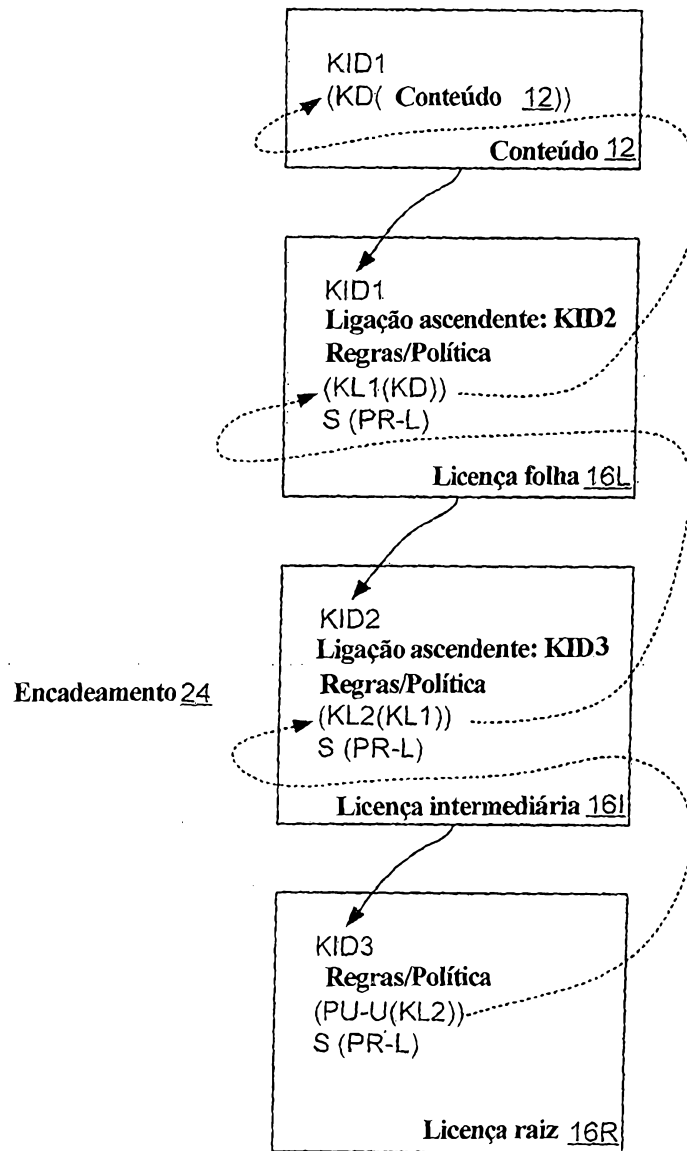


FIG. 4

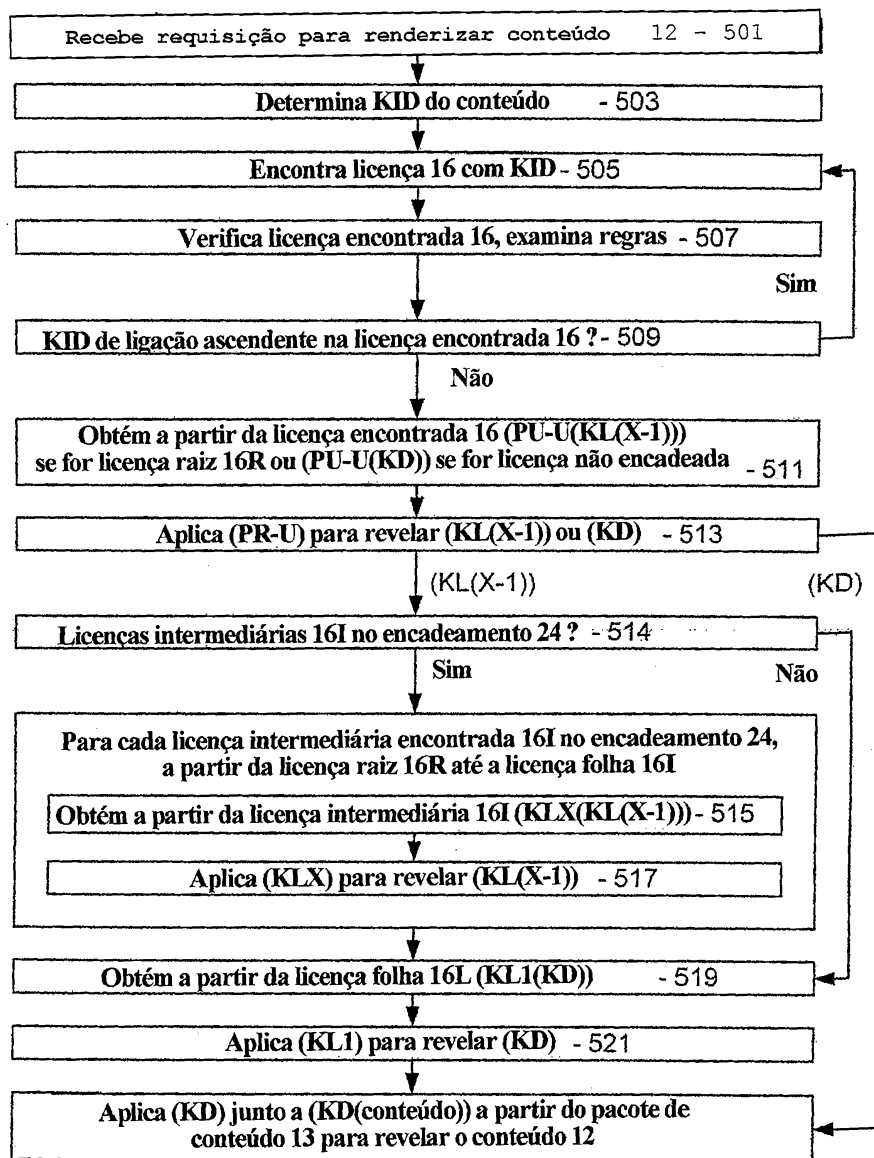


FIG. 5