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

(21) **PI 0610402-9 A2**

* B R P I 0 6 1 0 4 0 2 A 2 *

(22) Data de Depósito: 24/04/2006
(43) Data da Publicação: 10/01/2012
(RPI 2140)

(51) Int.Cl.:
H04L 9/08
H04H 60/23

(54) Título: MÉTODO, RECEPTOR E PRODUTO DE PROGRAMA PARA GERAÇÃO DO GRUPO DE CHAVE

(30) Prioridade Unionista: 25/04/2005 US 60/674959

(73) Titular(es): NOKIA CORPORATION

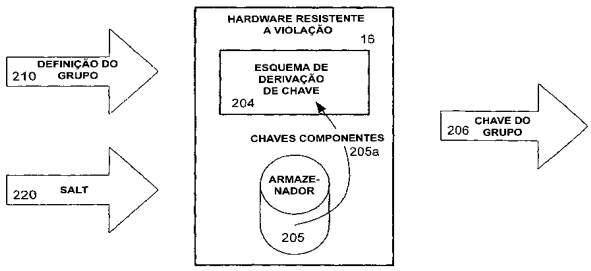
(72) Inventor(es): LAURI TARKKALA

(74) Procurador(es): Araripe & Associados

(86) Pedido Internacional: PCT IB2006000992 de 24/04/2006

(87) Publicação Internacional: WO 2006/114684de 02/11/2006

(57) Resumo: MÉTODO, RECEPTOR E PRODUTO DE PROGRAMA PARA GERAÇÃO DO GRUPO DE CHAVE. Um sistema de geração de chave é descrito, o qual provê a geração do grupo de chaves privilegiado baseado na entrada do grupo privilegiado, O sistema que executa a geração de chave possui componentes chaves armazenados correspondendo a cada sub-grupo X possível do grupo unitário, onde os sub-grupos X têm k ou menos membros. O grupo de chave privilegiado é gerado para o grupo privilegiado ao passar os componentes chaves ordenados dos sub-grupos X que não contêm os membros do grupo privilegiado para a função pseudo randômica.



"MÉTODO, RECEPTOR E PRODUTO DE PROGRAMA PARA GERAÇÃO DO GRUPO DE CHAVE".

CAMPO DA INVENÇÃO

A presente invenção se relaciona em geral ao campo da segurança e da criptografia. Esta invenção mais especificamente se relaciona à distribuição da chave nos sistemas de entrega de conteúdo.

DESCRIÇÃO DA TÉCNICA ANTERIOR

A. Fiat e M. Naor, Broadcast Encryption, Advances in Cryptology - CRYPTO'93 Proceedings, Lecture Notes in Computer Science, Vol. 773, 1994,páginas. 480- 491.

NIST. FIPS- 197: Advanced Encryption Standard.

http://csrc.nist.gov/publications/fips/fips 197/fips-197.pdf.

S. Frankel. AES-XCBC-MAC-96 Algorithm And Its Use With IPSec.

http://www.ietf.org/rfc/rfc3566.txt

NIST. FIPS-81 : DES Modes Of Operation.

http://www.itl.nist.gov/fipspubs/fip81.htm

H. Krawzyk. RFC 2104 - Keyed-Hashing for Message Authentication.

http://www.faqs.org/rfcs/rfc2104.html

NIST. FIPS 180-1 : Secure Hash Standard.

RESUMO DA INVENÇÃO

Um aspecto da invenção descrita provê um método de geração de um grupo chave para um conjunto de receptores dos usuários autorizados. O método provê um componente chave para cada subconjunto X possível de receptores autorizados com um número menor que k membros, onde k é uma constante predefinida. Uma função ordenadora injetora aloca os subconjuntos X em uma ordem determinada. Para o subconjunto de receptores autorizados, os subconjuntos X que não contêm os membros do subconjunto privilegiado são determinados. Os componentes da chave associados com

cada subconjunto X são identificados. Uma função pseudo randômica que aceita como entradas as chaves componentes associadas com subconjuntos segregados do conjunto privilegiado com um tamanho menor que k, conforme definido pela função ordenadora injetora, e produz um grupo chave.

Um outro aspecto da invenção descrita provê um receptor com um ambiente a prova de violação que executa a geração de grupo chave. O ambiente a prova de violação armazena uma variedade de componentes chave e uma ID do dispositivo. Para cada dispositivo, há pelo menos um componente chave armazenado correspondendo a cada subconjunto X possível do qual o dispositivo não é um membro, onde os subconjuntos X identificam cada conjunto de receptores com um número de membros menor que k. No recebimento da definição de um grupo privilegiado, o hardware do receptor a prova de violação determina se o receptor é membro do grupo privilegiado. Se sim, a lógica do ambiente a prova de violação determina os subconjuntos X de tamanho menor que k que não contém os membros do grupo privilegiado, estes grupos são ordenados conforme determinado pela função ordenadora injetora. Os componentes chave associados com os grupos ordenados são usados como parâmetros por uma função pseudo randômica determinada pela função ordenadora. A resposta da função pseudo randômica é a chave do grupo privilegiado.

BREVE DESCRIÇÃO DAS FIGURAS

Figura 1 – é um exemplo do sistema de distribuição do conteúdo no contexto dos sistemas e métodos descritos.

Figura 2 – é um exemplo do sistema de derivação de chave do receptor.

Figura 3 – é um exemplo de uma função mix() baseada na AES-XCBC-MAC.

Figura 4 – é um exemplo de uma função mix() baseada no HMAC_SHA1.

Figura 5 – é um exemplo de uma função mix() com chave de

comprimento variável baseado no HMAC_SHA1.

DESCRIÇÃO DETALHADA DA INVENÇÃO

Em um sistema de entrega de conteúdo, conforme mostrado na fig. 1, um provedor de conteúdo 10 transmite o conteúdo para um ou mais receptores 15 via um meio de transmissão. Um exemplo de tal sistema de entrega de conteúdo é a difusão de televisão via transmissão aérea, cabo, difusão de vídeo digital (DVB), satélite, redes de protocolo Internet ou qualquer outro sistema de entrega de multimídia incluindo Difusão Digital de Multimídia (DMB) e MediaFLO™ . Naturalmente, inúmeros outros tipos de conteúdo e de meios de transmissão adaptar-se-ão também a este modelo de entrega de conteúdo e se ajustaram ao contexto da invenção. Outros exemplos de tipos de conteúdo que poderiam ser distribuídos via este modelo incluem áudio, texto, vídeo games ou mídia interativa. Outros exemplos de meios de transmissão adequados incluem radiodifusão, celular, Bluetooth, IEEE802.11x, redes em malhas e WANs ou LAN com fio ou fibra ótica.

Provedores de conteúdo freqüentemente fornecem uma variedade de serviços aos seus usuários. Isto permite aos usuários formatar os serviços que recebem, de modo a atender às suas necessidades individuais. No contexto dos serviços de televisão, por exemplo, os usuários podem escolher entre canais premium, eventos pagar-para-ver (pay-per-view) e uma solicitação de programação especial. Para facilitar esta variedade, os provedores de conteúdo tipicamente criptografam alguns ou todos os seus conteúdos e somente permitem receptores autorizados a descriptografar os conteúdos correspondentes aos serviços que os usuários compraram.

Consistente com o sistema de criptografia, os provedores de conteúdo 10 empregarão hardware e software para criptografar pelo menos alguns dos conteúdos transmitidos e os receptores 15 terão hardware e software para descriptografar o conteúdo. O hardware dos receptores poderia ser incorporado em uma grande variedade de dispositivos, por exemplo, uma caixa seletora para televisão, um terminal móvel ou um computador de uso

geral. Para manter a segurança do esquema criptografado, o hardware e/ou software do receptor incluirá um ambiente à prova de violação 16 que contém a informação e a lógica exigida para participar do sistema de criptografia. O ambiente à prova de violação 16 ajuda a garantir que os usuários tentem quebrar o sistema de criptografia não tendo acesso aos segredos do sistema. O ambiente à prova de violação 16 pode ser incorporado via quaisquer dos sistemas e métodos conhecidos na técnica.

O gerenciamento do sistema de criptografia/descriptografia, contudo, cria algumas dificuldades. Um problema particular é o gerenciamento e distribuição das chaves secretas e algorItmos usados para operar o sistema. Como o número de receptores no sistema ou o número de eventos discretos de criptografia se tornam grandes o gerenciamento das chaves se torna extremamente problemático.

Os sistemas e métodos descritos fornecem eficiente e segura geração e distribuição das chaves necessárias para criptografar e descriptografar o conteúdo. Os sistemas e métodos descritos permitem a ambos, provedor do conteúdo e ambiente à prova de violação dos receptores autorizados 16 para gerar chaves associativas a partir de um grupo de informação da secreta compartilhada e lógica. Além disso, o sistema descrito permite ao provedor de conteúdo e ao ambiente à prova de violação dos receptores 16 gerar combinações de grupos de chaves para um subgrupo de usuários autorizados. A definição de grupo autorizado permite que o provedor de conteúdo limite o número de eventos de criptografia e também limite a quantidade de informação transmitida, desse modo, reforçando a segurança do sistema.

Especificamente, os sistemas e métodos descritos fornecem a derivação do grupo de chaves em um ambiente de difusão, onde o grupo de chaves não revela a informação sobre os segredos armazenados no ambiente à prova de violação dos receptores 16. O ambiente à prova de violação 16 é necessário para implementar o esquema de derivação de chave

e armazenar chaves componentes. As chaves componentes são chaves de segurança armazenadas no ambiente à prova de violação do receptor 16, que poderiam ter sido colocadas no receptor antes da distribuição do dispositivo para o usuário. Preferivelmente, cada ambiente à prova de violação apenas armazena as chaves necessárias para gerar a chave grupo para os grupos autorizados do qual o receptor é um membro. O ambiente à prova de violação não necessita armazenar as chaves usadas para gerar grupos de chaves dos quais ele não é um membro.

Por exemplo, como mostrado na FIG. 2, um ambiente à prova de violação do receptor 16, inclui um esquema de derivação de chave 204 e armazenamento seguro 205, o qual armazena as chaves componentes 205a.

Para gerar um grupo particular de chaves 206, o ambiente à prova de violação 16 aceita como entrada uma definição de grupo 210 e, opcionalmente, salt 220, que é, por exemplo, uma constante global, específica para uma certa definição de grupo, hora do dia ou algum outro parâmetro independente das chaves componentes. Para garantir que a integridade do sistema de criptografia seja mantida, um ambiente à prova de violação do receptor 16 somente produzirá uma chave grupo se o usuário for um membro do grupo. Isto exige que a ID do dispositivo seja armazenada em um dispositivo de armazenamento seguro, de modo que o receptor possa reconhecer quando ele é um membro do grupo fornecido na definição de grupo 210. Vantajosamente, mesmo se algumas das chaves de grupo derivadas forem expostas aos usuários que tentam violar o sistema de criptografia, as secretas de longo prazo no ambiente à prova de violação 16 permanecem seguras. Além disso, o risco da exposição da chave grupo pode ser mitigado pela modificação freqüente dos parâmetros salt. Adicionalmente, um receptor que não é membro de um grupo autorizado não será capaz de violar a chave grupo utilizando o método descrito porque ele não disporá dos parâmetros necessários. A proteção dos sistemas, desse modo, não é baseada somente na determinação do ambiente à prova de violação quer

este seja um membro do grupo autorizado.

Para o objetivo desta discussão, o grupo **U** é assumido para ser o grupo de todos os usuários. Naturalmente, na implementação de um sistema completo, o provedor de conteúdo pode operar múltiplos domínios independentes **U**. Assumiremos **n** = |**U**| para ser o tamanho do grupo. O provedor de conteúdo seleciona um valor de **k** que define a resistência do sistema, onde **k<n.** Esta resistência define a quantidade mínima do número de usuários que devem violar o ambiente à prova de violação e cooperar entre si para violar o esquema de criptografia. A seleção de **k** é uma decisão de projeto. Um valor grande para **k** conduz a um número maior de chaves, mas resulta em um sistema de criptografia que é mais difícil de violar. Por outro lado, um valor pequeno de **k** resulta em um sistema menos robusto, mas requer um número relativamente menor de chaves. Por exemplo, se **k** foi estabelecido para 2, o sistema seria seguro considerando que o ambiente à prova de violação permanecesse seguro, mas se dois usuários obtivessem as secretas no ambiente à prova de violação, eles poderiam juntar e invadir o sistema.

Uma primeira função ordenadora injetora **f** que transforma os membros do grupo **U** em membros do grupo **Z** , i.e., **f : U → Z**, de tal modo que os membros de **U** estão ordenados em **Z**. Além disso, para dois membros **a,b** de **U**, **a<b** se e somente se **f(a)<f(b)**. Uma outra função ordenadora injetora **g(X)** é definida para ordenar os subgrupos de **U.** Um exemplo de tal função é **g(X)** $= \sum_{u \in X} 2^{f(u)}$. Quaisquer outras funções que forneçam a ordenação injetora para os subgrupos **U**, contudo, podem ser usadas e seriam prontamente visualizadas na técnica. Uma chave é destinada para cada possível grupo **X** no grupo **U** para o qual |**X**| **< k.** Uma chave **K_i** é atribuída onde **i=g(X)**. Em um projeto alternativo, toda a descrição relativa as chaves cria parte de uma chave onde o restante da chave é criado usando um outro procedimento.

Para cada dispositivo, o ambiente à prova de violação apenas

armazena as chaves **K_i** que correspondem aos subgrupos de **U** com tamanho menor que **k** do qual ele não é um membro. Isto (juntamente com a derivação de chave abaixo descrita), implica que menos que **k** membros de **U** são incapazes de calcular a chave de grupo dos grupos dos quais não sejam membros.

O grupo de usuários autorizados é definido como **Y**, o qual é subgrupo de **U** contendo os receptores autorizados. **Y** serve como a definição de grupo 210 enviada pelo provedor de conteúdo. Alternativamente, o grupo de usuários não incluídos em **Y** poderia servir como uma definição de grupo. Para um grupo autorizado **Y**, a chave do grupo 206 é gerada pelo emprego de uma função pseudo randômica que pode assumir um número arbitrário de entradas de comprimentos arbitrários, no contexto da descrição denominada de mix(). Para um dado grupo **Y**, os parâmetros para mix() são derivados de cada subgrupo **X** para o qual $|X| < k$ que não inclui os membros de **Y**, i.e., **U – Y**. Cada subgrupo é um membro de **X** e, então, tem uma chave associada **K_i** armazenada por cada receptor. As chaves **K_i** para cada **X** de **U – Y** são usadas como parâmetros para a função mix(). Além do mais, elas são usadas na ordem definida por **g(X)**. Adicionalmente às chaves ordenadas **K_i**, o parâmetro salt, conforme discutido acima, pode ser opcionalmente enviado em conjunto com a definição de grupo e adicionado como parâmetro para mix().

Conforme mencionado, o parâmetro salt pode ser enviado como um parâmetro separado da definição de grupo. Este salt pode ser solicitado para que seja de um determinado formato (por exemplo, exatamente **m** bits em comprimento ou no máximo **m** bits no comprimento). Desse modo, se o salt não satisfizer os critérios estabelecidos, então a derivação da chave do grupo falha, então, necessita prover segurança adicional.

Três exemplos de implementação de função mix() são descritos a seguir, dois baseados na HMAC_SHA1 e um na AES_XCBC_MAC. Para as descrições apresentadas, o operador binário || é usado para descrever a

concatenação. Naturalmente, inúmeras outras implementações e exemplos de funções mix() adequadas poderiam ser imediatamente visualizadas sem sair do conceito inventivo da invenção.

EXEMPLO FUNÇÃO MIX() BASEADA NA AES-XCBC-MAC

Esta seção descreve uma função mix() baseada na AES-XCBC-MAC como descrito por S. Frankel, citado acima, Modo-Contador e modo de Realimentação Cifrado conforme descrito no NIST FIPS-81: DES Modos de Operação, citados acima. AES-CBC-MAC é usada pela criação de um código de autenticação da mensagem (MAC) empregando AES no modo CBC conforme descrito na NIST.FIPS-81.

A função pseudo randômica é definida considerando os parâmetros (**k,x,j**) baseados na AES-XCBC-MAC que fornece até **j** blocos AES como necessário. A entrada da função é uma chave AES, uma cadeia de bits **x** de blocos AES. Os blocos são denominados **x_1, x_2, ...**

AES_**k(x)** é usada para denominar criptografia com AES sobre um bloco **x** de planilha de texto simples usando chave **k**.

AES_CBC_MAC_k(**x**) é usada para indicar processamento de uma MAC no modo-CBC usando AES com chave **k** sobre os blocos **x** da planilha de texto. A entrada é assumida para ser de comprimento adequado (i.e. um múltiplo do tamanho do bloco AES).

A função pseudo randômica é processada como a seguir:

1. Seja k1 = AES_k (P1).
2. Seja k2 = AES k (P2).
3. C_1= AES_k1(AES_CBC_MAC_k1(x) XOR k2 XOR 0x01)
4- Para cnt = 2 a j

C_cnt = AES_k1(AES_CBC_MAC_k1(x II C_{cnt - 1}) XOR k2XOR cnt)

A função pseudo randômica sempre cria até j AES blocos de dados. A função mix(salt, k_1,.....,k_m) é agora definida como :

1. T_1 = função pseudo randômica (k_1, SALT,,j)

2. Para cnt = 2 a m

a. T_cnt = função pseudo randômica (k_cnt,T_{cnt - 1}, j)

As constantes P1 e P2 podem ser definidas à vontade, observando que P1! = P2. Pode-se por exemplo usar os valores P1 = 0x01010101010101010101010101010101 e P2 = 0x02020202020202020202020202020202.

Esta função mix() resulta em uma seqüência de bits T_m (onde m é o número de chaves na entrada) que é a chave para o grupo autorizado.

A FIG. 3 Ilustra a implementação de um exemplo de função mix baseada na AES-XCBC-MAC para o caso i=1 e cnt >1 e o salt tem o comprimento exato de um bloco AES. K_i 301 é aplicado aos blocos AES 302, 303 e 304. A constante P1 305, definida tal que P1!=P2, juntamente com a entrada da chave 301 são aplicadas ao bloco AES 302. A saída do bloco AES 302 juntamente com salt 309 são aplicados ao bloco AES 307. The XOR 311 de T_i, {j - 1} 310 e a saída do bloco AES 307 é aplicada ao bloco AES 308 juntamente com a saída do bloco AES 302. A chave 301 e a constante P2 306 são aplicadas ao bloco AES 303. A XOR 313 de j 312, como acima definido, e a saída do bloco AES 303 são aplicadas ao XOR 314 juntamente com a saída do bloco AES 308. A saída de XOR 314 e a chave 301 são aplicadas no bloco AES 304 para produzir T_i,j 315. O procedimento resulta no bloco T_i,j. Se somente uma chave é membro do grupo privilegiado e j=2, então este bloco seria o segundo bloco da chave do grupo de saída.

EXEMPLO FUNÇÃO MIX() BASEADA NO HMAC_SHA1

A função mix() baseada no HMAC_SHA1 é algo mais simples do que a implementação descrita acima. A função pseudo randômica considerando os parâmetros **(k,x,j)** que fornecem as saídas dos blocos **j** SHA 1, conforme descrito na NIST.FIPS 180-1: Secure Hash Standard, blocos de dados (160-bits) dada uma chave **k** e uma seqüência de bits **x**. Denominaremos por HMAC_SHA1 **(k,x)** um HMAC_SHA1 processado usando uma chave **k** e uma seqüência de bits de entrada **x**. A função pseudo

randômica é como a seguir:

1. C_1 = HMAC_SHA1(k,x II 0x01)

2. Para cnt = 2 a j

C_cnt = HMAC_SHA1(k,x II C_{cnt - 1} II cnt)

A função pseudo randômica sempre cria até **j** blocos de dados SHA1 (160 bits). A função mix(**salt, k_1,...,k_m**) agora é definida como:

1. T_1 = pfr(**k_1, salt, j**)

2. Para cnt = 2 a m

T_cnt = pfr(k_cnt, T_ {cnt -1}, j)

Este procedimento mix() resulta em uma seqüência de bits T_m (onde m é o número de chaves na entrada) isto é, a chave para o grupo privilegiado.

A FIG. 4 Ilustra a função mix() baseada no HMAC_SHA1. C_i,0 é considerado como sendo uma seqüência vazia. O XOR 404 do Ipad 405 e a interação de K_i 403 é aplicada juntamente com Salt na concatenação com C_i, { j-1} concatenado com j 401 para garantir Secure Hash Algorithm1 (SHA1) 402. O XOR 407 do Opad 405 e a interação de K_i 403 está aplicado juntamente com a saída da SHA1 402 para SHA1 408 para criar C_i,j 409. Este procedimento produz um bloco de saída SHA1. A iteração para cada valor de i ao longo de todos os blocos requeridos, de 1 até j, produz uma seqüência de blocos C_mj (onde m é o número de chaves de entrada) que quando concatenada produz a chave do grupo.

EXEMPLO DE FUNÇÃO MIX() BASEADA NO HMAC_SHA1 COM CHAVE DE COMPRIMENTO VARIÁVEL

O uso de chaves de comprimentos variáveis com o HMAC_sha1 simplifica e acelera de forma significativa a função mix(). A função mix(salt, k_1, ..., k_n) então é processada como a seguir:

1. T_1 = HMAC_SHA1(K_1 II ...II K_n, salt II 0x01)

2 Para cnt = de 2 a j

T_cnt = HMAC_SHA1(K_1 ii ... II K_n, salt II T_{ cnt -1} II cnt)

Nesta função todas as chaves são concatenadas juntas na ordem definida pela função ordenadora injetora. A realimentação cifrada e o modo contador são combinados e processados por HMAC_SHA1 sobre o salt.

A FIG.5 ilustra a função mix() baseada na chave de comprimento variável HMAC_SHA1. T_0 é considerada como sendo uma seqüência vazia. O XOR 504 do Ipad 505 e K_1 II ... II K_n 503 é aplicado juntamente com salt em concatenação com T_ {j - 1} concatenado com T_ {j - 1}, concatenada com j 501 para garantir Secure Hash Algorithm1 (SHA1) 502. O XOR 507 do Opad 505 e a concatenação de K_1 II ... II K_n 503 é aplicada juntamente com a saída da SHA1 502 na SHA1 508 para criar T_j 509. Para a função descrita, no caso em que j>1 (se j=1 então o bloco anterior deveria ser omitido). Ipad e Opad são novamente constantes de comprimento igual à concatenação das chaves definidas em H. Krawzyk. RFC 2104 – Keyed-Hashing for Message Authentication.

Várias características e vantagens da presente invenção são aparentes a partir da especificação detalhada, e então, é pretendido que as reivindicações em anexo, cubram todas as características e vantagens da invenção, as quais se encaixam dentro do verdadeiro conceito inventivo e escopo da invenção.

Ademais, uma vez que inúmeras modificações e variações imediatamente poderão ocorrer àqueles familiarizados com a técnica, não é desejável que a presente invenção seja limitada às exatas instruções e operações aqui ilustradas e descritas. Assim sendo, todas as modificações e equivalentes adequadas que possam ser delas extraídas são pretendidas para estar dentro do escopo das reivindicações.

REIVINDICAÇÕES

1. Método de geração do grupo de chave **CARACTERIZADO** pelo fato de que compreende:

- um grupo de receptores, que provê um componente de chave para cada sub-grupo X possível de receptores com menos do que k membros, onde k é uma constante pré-definida;

- definir uma função de ordenação iterativa que ordena os sub-grupos X;

- para um sub-grupo de receptores privilegiados, determinar quais sub-grupos X não contêm os membros do sub-grupo privilegiado e identificar os componentes chaves com cada sub-grupo X;

- definir uma função pseudo randômica que carrega um número arbitrário dos componentes de chave como entradas e saídas do grupo de chave; e

- usar os componentes de chave associados com os sub-grupos de X de tamanho menor do que k que não contêm os membros do sub-grupo privilegiado como entrada para a função pseudo randômica, onde os componentes chaves são aplicados à função pseudo randômica de forma a determinada pela função de ordenação iterativa e a saída da função pseudo randômica é um grupo de chave específico do receptor privilegiado.

2. Método de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que também compreende uma função de ordenação iterativa adicional que designa os componentes chave para cada sub-grupo X.

3. Método de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que é executado pelo receptor, e onde o receptor executa apenas o método se este for membro do sub-grupo privilegiado.

4. Método de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que a função pseudo randômica é baseada no AES-XCBC-MAC.

5. Método de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que a função pseudo randômica é baseada no HMAC_SHA1.

6. Método de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que a função pseudo randômica carrega o parâmetro adicional salt.

7. Receptor **CARACTERIZADO** pelo fato de que compreende:

- um ambiente a prova de violação, compreendendo armazenagem e lógica;

- uma pluralidade de componentes chaves e a ID do dispositivo armazenada em um ambiente a prova de violação, onde há ao menos um componente chave correspondendo a cada sub-grupo X possível do qual o receptor não é membro, onde os sub-grupos X descrevem cada grupo de receptores com menos do que k membros;

- onde no recebimento de uma definição de grupo privilegiada, a lógica no ambiente a prova de violação determina os sub-grupos X que não contêm os membros do grupo privilegiado, cada grupo é ordenado como determinado por uma função de ordenação iterativa, os componentes chave associados com os grupos ordenados são usados como parâmetros para a função pseudo randômica e aplicados na ordem ditada pela função de ordenação; e

- onde a saída da função pseudo randômica é um grupo de chave privilegiado.

8. Receptor de acordo com a reivindicação 7, **CARACTERIZADO** pelo fato de que a função pseudo randômica é baseada no AES-XCBC-MAC.

9. Receptor de acordo com a reivindicação 7, **CARACTERIZADO** pelo fato de que a função pseudo randômica é baseada no HMAC_SHA1.

10. Receptor de acordo com a reivindicação 7, **CARACTERIZADO** pelo fato de que a função pseudo randômica carrega o parâmetro adicional salt.

11. Produto de programa para a geração do grupo de chave **CARACTERIZADO** pelo fato de que compreende:

- um dispositivo legível de computador;

- um código de programa armazenado no meio legível de computador que define uma função de ordenação iterativa que ordena os sub-grupos X, onde os sub-grupos X são sub-grupos do grupo de todos os receptores possuindo

menos do que um número de membros pré-determinado;

- um código de programa armazenado no meio legível de computador que no recebimento da definição do grupo determina quais sub-grupos X não contêm quaisquer membros de dentro da definição do grupo e identifica um componente chave associado com cada sub-grupo X;

- um código de programa armazenado no meio legível de computador compreendendo uma função pseudo randômica que carrega um número arbitrário de componentes chave como entradas e saídas do grupo de chave; e

- um código de programa armazenado no meio legível de computador que usa os componentes chaves associados com os sub-grupos de X que não contêm os membros dentro da definição do grupo como entradas para a função pseudo randômica, onde os componentes chaves são aplicados à função pseudo randômica de forma determinada pela função de ordenação iterativa e a saída da função pseudo randômica é um grupo de chave específico do receptor privilegiado.

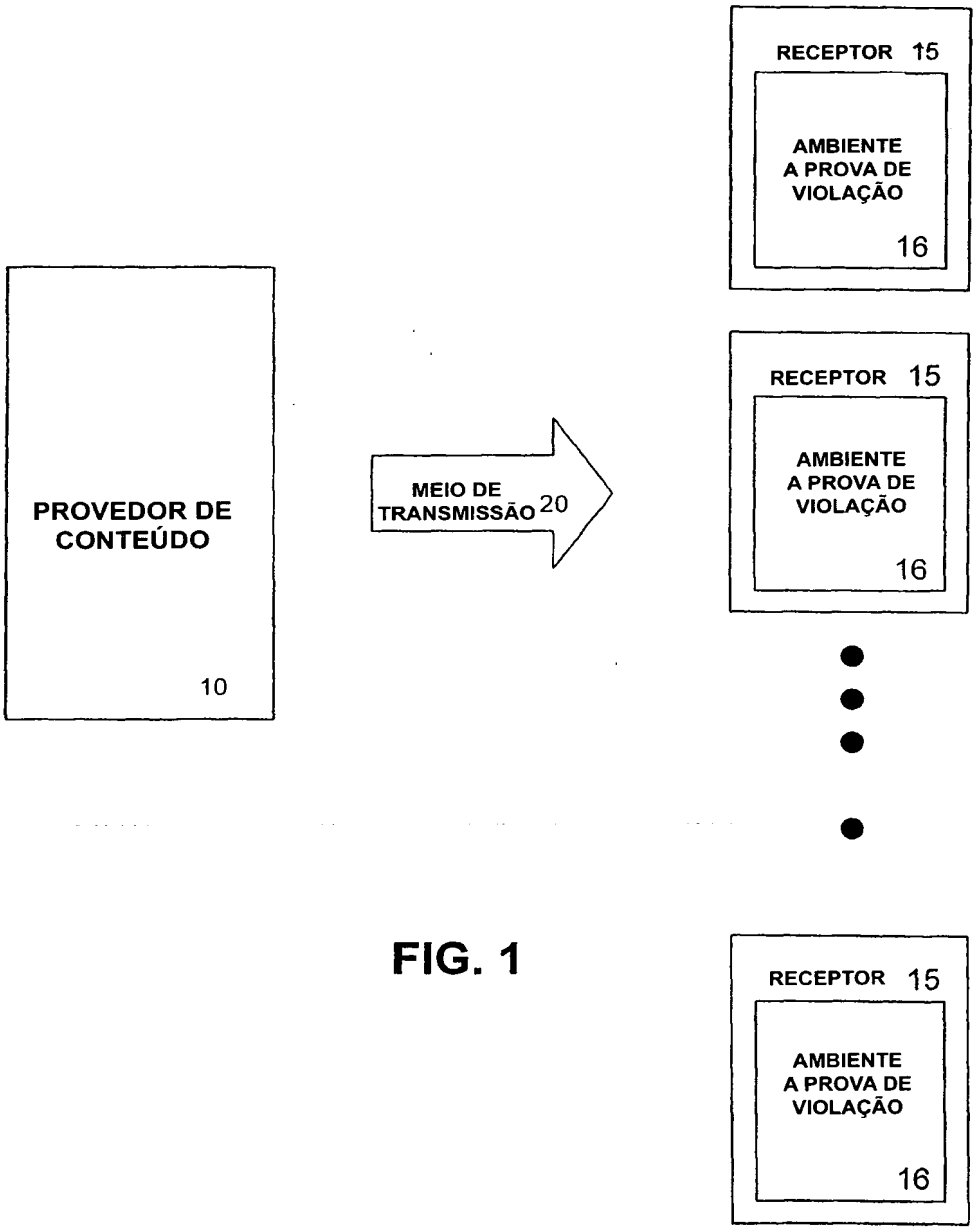

PROVEDOR DE CONTEÚDO
10
MEIO DE TRANSMISSÃO 20
RECEPTOR 15
AMBIENTE A PROVA DE VIOLAÇÃO
16
RECEPTOR 15
AMBIENTE A PROVA DE VIOLAÇÃO
16
RECEPTOR 15
AMBIENTE A PROVA DE VIOLAÇÃO
16

FIG. 1

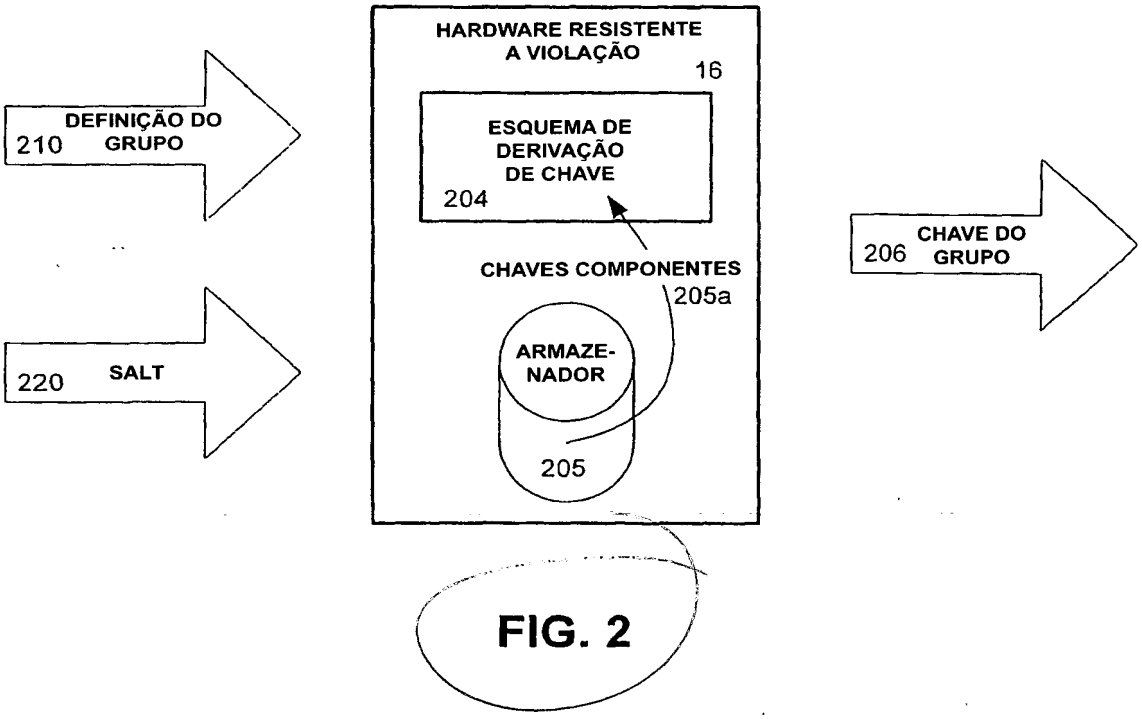

HARDWARE RESISTENTE
A VIOLAÇÃO
16
DEFINIÇÃO DO
210 GRUPO
ESQUEMA DE
DERIVAÇÃO
DE CHAVE
204
CHAVES COMPONENTES
205a
ARMAZE-
NADOR
205
220 SALT
206 CHAVE DO
GRUPO

FIG. 2

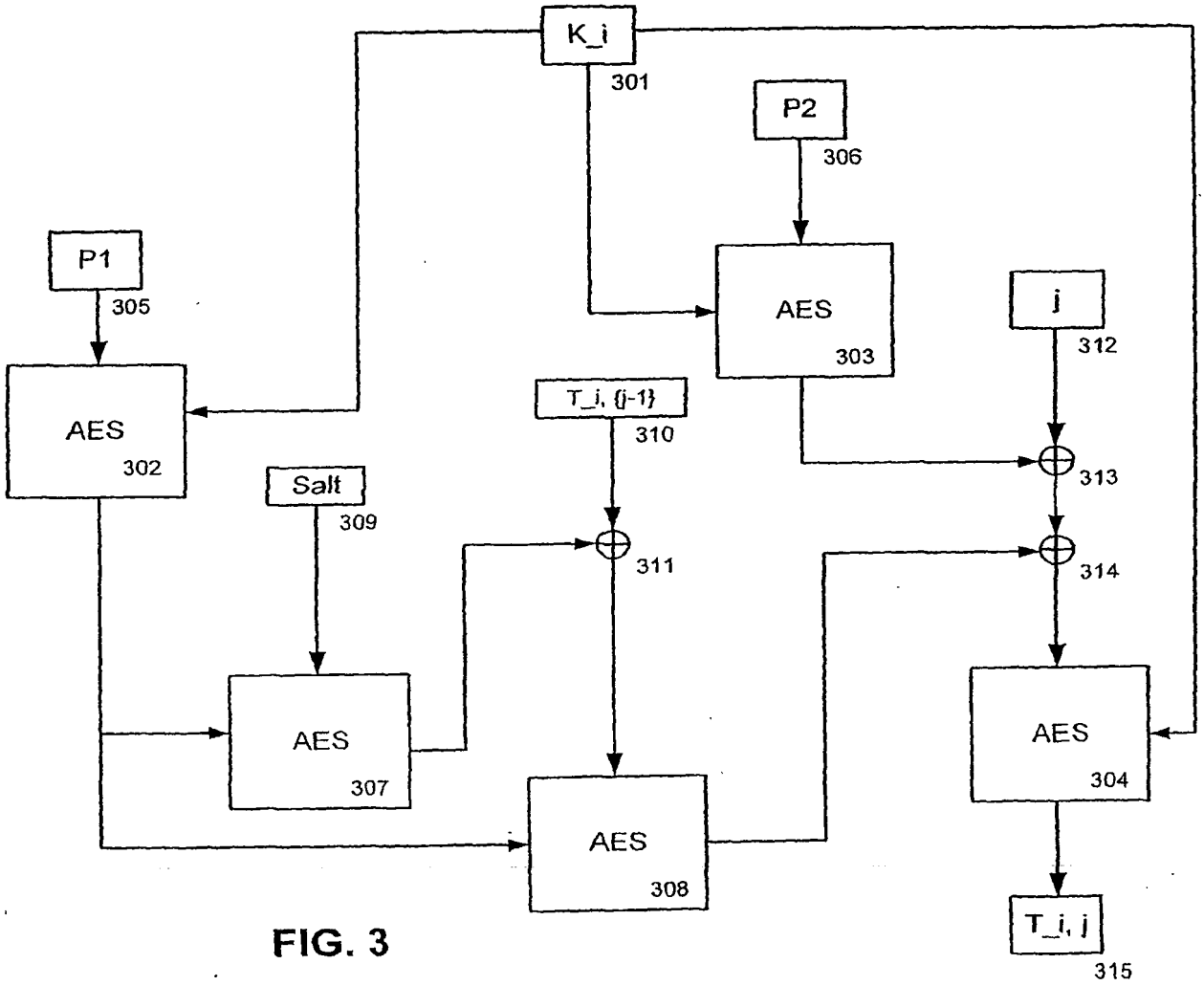

K_i
301
P2
306
P1
305
AES
303
j
312
AES
302
T_i, {j-1}
310
Salt
309
313
311
314
AES
307
AES
304
AES
308
T_i, j
315

FIG. 3

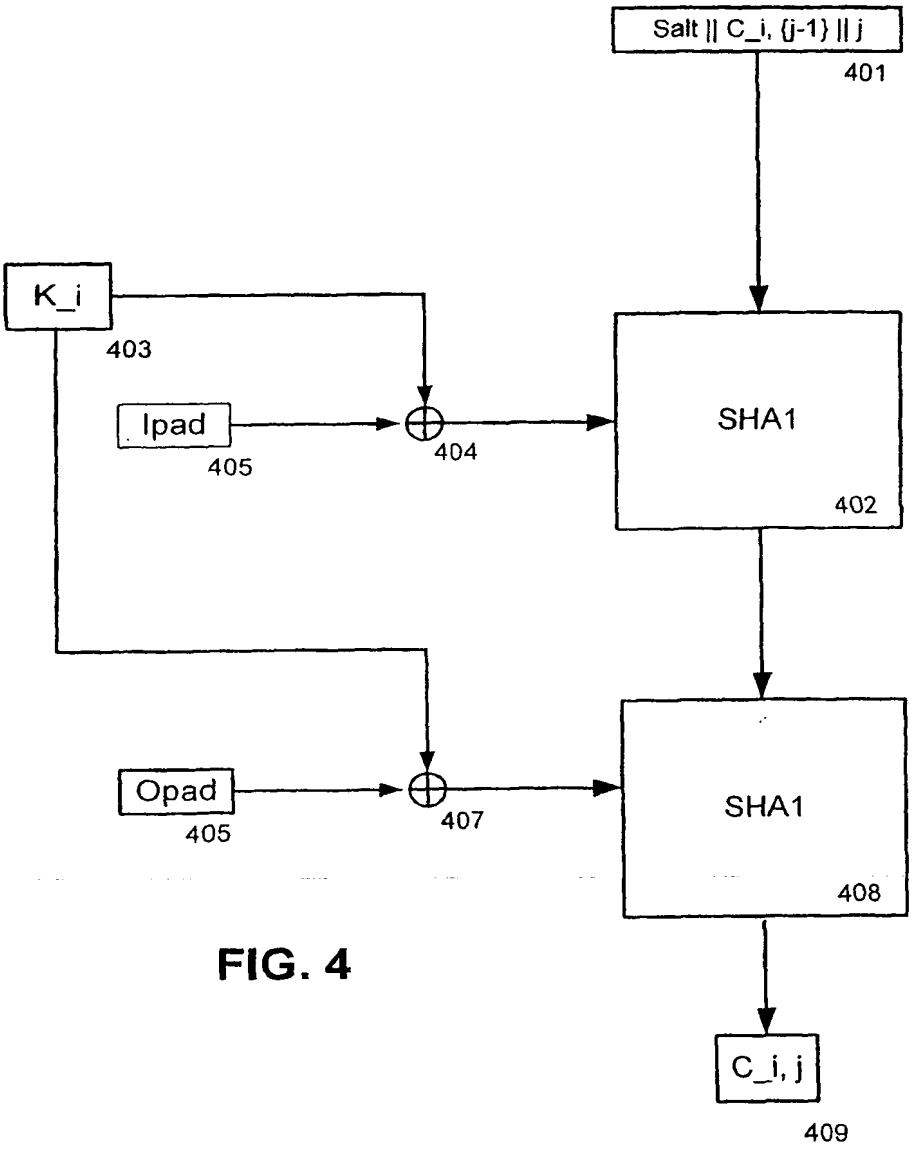

Salt || C_i, (j-1) || j
401
K_i
403
Ipad
405
404
SHA1
402
Opad
405
407
SHA1
408
C_i, j
409

FIG. 4

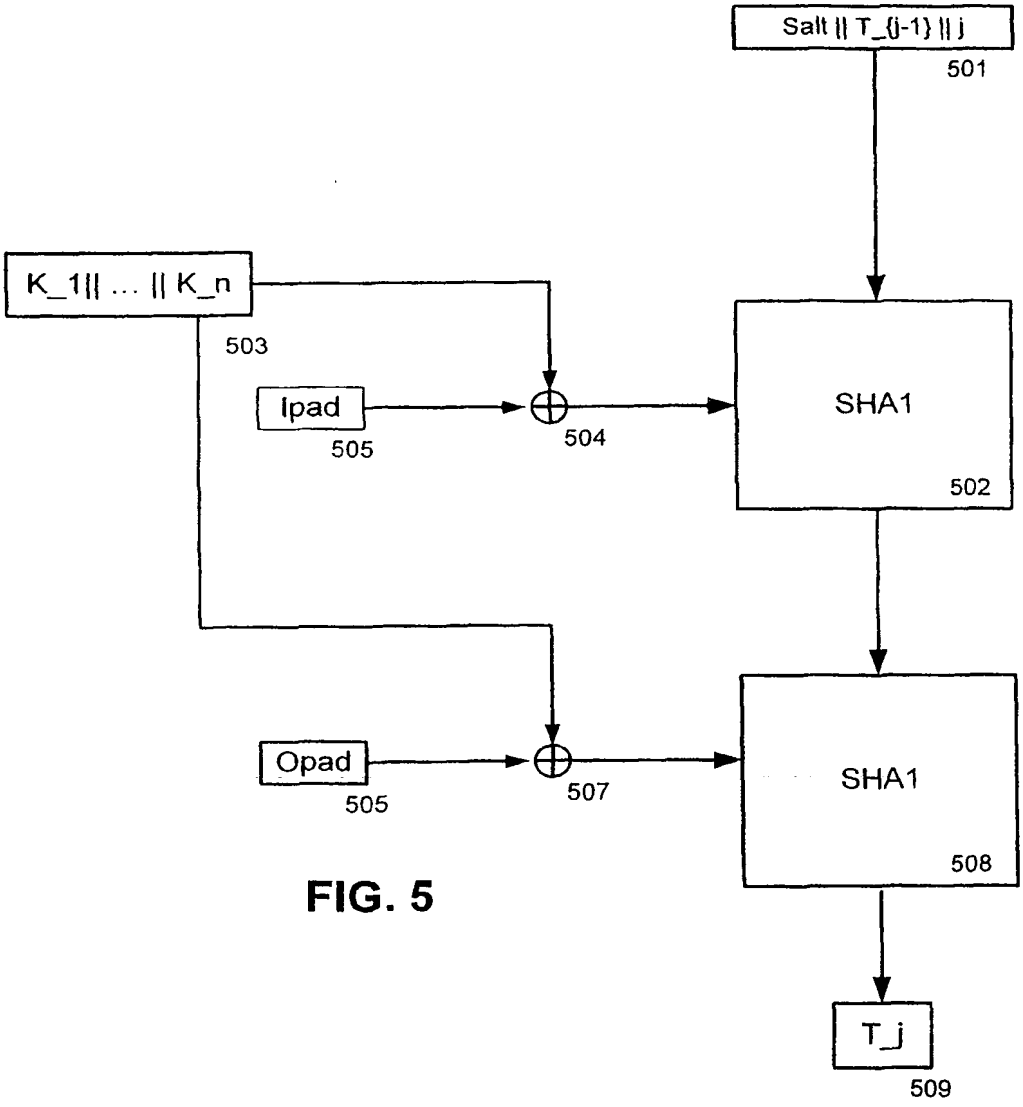

Salt || T_{i-1} || j
501
K_1|| ... || K_n
503
Ipad
505
504
SHA1
502
Opad
505
507
SHA1
508
T_j
509

FIG. 5

RESUMO

"MÉTODO, RECEPTOR E PRODUTO DE PROGRAMA PARA GERAÇÃO DO GRUPO DE CHAVE."

Um sistema de geração de chave é descrito, o qual provê a geração do grupo de chaves privilegiado baseado na entrada do grupo privilegiado. O sistema que executa a geração de chave possui componentes chaves armazenados correspondendo a cada sub-grupo X possível do grupo unitário, onde os sub-grupos X têm k ou menos membros. O grupo de chave privilegiado é gerado para o grupo privilegiado ao passar os componentes chaves ordenados dos sub-grupos X que não contêm os membros do grupo privilegiado para a função pseudo randômica.