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

**(21) BR 112020007373-3 A2**

**(22) Data do Depósito:** 12/10/2018

**(43) Data da Publicação Nacional:** 29/09/2020

**(54) Título:** PRIMEIRO NÓ DE REDE ELETRÔNICA CONFIGURADO PARA UMA OPERAÇÃO CRIPTOGRÁFICA, MÉTODO DE OPERAÇÃO CRIPTOGRÁFICA ELETRÔNICA PARA UM PRIMEIRO NÓ DE REDE ELETRÔNICA E MÍDIA LEGÍVEL POR COMPUTADOR

**(51) Int. Cl.:** H04L 9/08; H04L 9/30.

**(30) Prioridade Unionista:** 17/10/2017 EP 17196926.4.

**(71) Depositante(es):** KONINKLIJKE PHILIPS N.V..

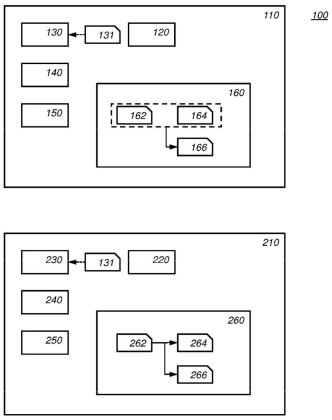
**(72) Inventor(es):** OSCAR GARCIA MORCHON; LUDOVICUS MARINUS GERARDUS MARIA TOLHUIZEN.

**(86) Pedido PCT:** PCT EP2018077823 de 12/10/2018

**(87) Publicação PCT:** WO WO/2019/076737 de 25/04/2019

**(85) Data da Fase Nacional:** 14/04/2020

**(57) Resumo:** Trata-se de um nó de rede eletrônica (110) configurado para uma operação criptográfica. O nó de rede obtém uma matriz compartilhada (A) selecionando-se números inteiros, polinômios e/ou coeficientes de polinômio a partir de um grupo compartilhado, em que o grupo compartilhado é compartilhado com o segundo nó de rede, em que a seleção é realizada de acordo com uma ou mais funções de seleção.



PRIMEIRO NÓ DE REDE ELETRÔNICA CONFIGURADO PARA UMA OPERAÇÃO CRIPTOGRÁFICA, MÉTODO DE OPERAÇÃO CRIPTOGRÁFICA ELETRÔNICA PARA UM PRIMEIRO NÓ DE REDE ELETRÔNICA E MÍDIA LEGÍVEL POR COMPUTADOR

Campo da invenção

[001] A invenção se refere a um nó de rede configurado para operação criptográfica, a um método de operação criptográfica e a uma mídia legível por computador.

Antecedentes da invenção

[002] Na criptografia, um protocolo de concordância de chaves é um protocolo segundo o qual duas ou mais partes, que ainda não compartilham uma chave comum, podem concordar a respeito dessa chave. De preferência, ambas as partes podem influenciar o resultado, de modo que nenhuma das partes possa forçar a escolha da chave. Um invasor que espiona toda a comunicação entre as duas partes não precisa ter nenhum conhecimento sobre a chave. Contudo, embora o invasor que observa a mesma comunicação saiba pouco ou nada sobre ela, as próprias partes podem derivar uma chave compartilhada. Protocolos de concordância de chaves são úteis, por exemplo, para proteger a comunicação, por exemplo, para criptografar e/ou autenticar mensagens entre as partes.

[003] A fim de facilitar a comunicação segura entre as partes, protocolos de concordância de chaves são, às vezes, ainda subdivididos em esquemas de troca de chaves criptográficas (KEX) e encapsulação de chave criptográfica (KEM). Os esquemas de encapsulação de chave criptográfica (KEM) usam criptografia assimétrica para estabelecer um segredo compartilhado entre duas partes, com o uso de um valor publicamente conhecido (por exemplo, chave pública) e de

propriedade secreta (por exemplo, chave secreta) para cada parte.

[004] Esquemas KEX envolvem a troca de chaves públicas por cada uma das partes, que são, então, independentemente usadas pela outra parte juntamente com sua própria chave secreta para calcular o segredo comum compartilhado. Um exemplo bem conhecido de um esquema KEX é a troca de chaves de Diffie-Hellman, mencionado acima, cuja segurança é baseada em resolver o problema de logaritmo discreto. Um recurso interessante de alguns esquemas KEX é que o real segredo final compartilhado nunca é trocado entre as partes, nem mesmo na forma criptografada, mas é calculado independentemente pelas duas partes em cada extremidade. Isso resulta em um recurso desejável conhecido como sigilo futuro, que garante que até mesmo o comprometimento da chave secreta de uma das partes a longo prazo por um invasor no futuro não comprometa o sigilo de mensagens criptografadas trocadas no passado.

[005] Esquemas KEM estabelecem um segredo compartilhado entre duas entidades ou partes com o uso de criptografia assimétrica por uma das partes, geralmente, o iniciador da comunicação, para criptografar (usando a chave pública da outra parte) e transmitir um segredo compartilhado para a outra parte, conhecida como o respondente, que pode, então, descriptografá-lo (usando sua chave secreta) e, então, usá-lo para comunicação segura com a parte iniciadora. Os esquemas KEM podem obter apenas sigilo futuro se forem efêmeros. De outro modo, qualquer invasor que comprometa a chave secreta de uma das partes para uma sessão anterior e que tenha registrado

todas as mensagens trocadas entre as partes nessa sessão pode recuperar o segredo compartilhado para aquela sessão específica.

[006] Devido às crescentes necessidades de segurança na Internet das coisas, os esquemas de troca de chaves também precisam alcançar alta eficiência (isto é, quantidade mínima de comunicação ou requisitos de largura de banda) enquanto também permanecem seguros contra adversários clássicos, bem como com capacidade quântica.

[007] Avanços recentes indicam que computadores quânticos podem ser viáveis daqui a alguns anos. Talvez em apenas 5 a 10 anos. Assim que um computador quântico estiver disponível, a maioria dos esquemas criptográficos de chave pública existentes se tornarão inseguros, uma vez que os problemas subjacentes, nos quais eles são baseados, podem ser eficientemente resolvidos com um computador quântico. Portanto, é muito importante projetar algoritmos de chave pública resistentes quânticos, isto é, algoritmos de chave pública que não possam ser rompidos por meio de um computador quântico.

[008] Há várias classes de algoritmos de resistência quântica: baseados em reticulados, baseados em código, baseados em isogenia etc. Este documento se refere a algoritmos de resistência quântica baseados em reticulados. Sistemas criptográficos baseados em reticulados são uma área ativa de pesquisa na qual muitos tipos diferentes de sistemas são propostos. Por exemplo, o reticulado pode ser definido em termos de uma matriz de números, por exemplo, elementos de algum campo finito ou com o uso de um polinômio etc. O tipo de operação criptográfica também pode variar. Por exemplo, a criptografia baseada em reticulado está disponível para: Encapsulação de

chave (KEX), Troca de chaves (KEX), Criptografia de chave pública (PKE), assinaturas digitais etc. Exemplos desses regimes são:

[009] KEX baseada em LEW, como Frodo

[010] KEX baseada em RLWE e CPA-KEM Newhope e NewHopeSimple.

[011] CPA-PKE baseado em reticulados, modulares CPA-KEM e CCA-KEM, como Kyber

[012] KEX baseado em LWR, como spKEX

[013] Cada um dos esquemas acima compartilha algumas características comuns. Cada um deles exige uma matriz compartilhada $A$, que é, em algum ponto, compartilhada com outro nó. A matriz compartilhada $A$ pode ser preenchida com números inteiros módulo um módulo $q$, ou com polinômios módulo um módulo $q$ e um polinômio de redução $f$. Os inventores perceberam que é recomendado não usar a matriz compartilhada com muita frequência. Por exemplo, no caso de protocolos KEX, um possível caso de usuário é usar uma matriz compartilhada particular apenas uma vez. No entanto, visto que o tamanho da matriz pode, por exemplo, ser 600 por 600, e o módulo $q$ poderia ser 14 bits, então, uma quantidade de dados na matriz compartilhada poderia ser significativa. Independentemente de a matriz ser comunicada ou gerada em ambos os lados, isso é uma sobrecarga importante. Especialmente para um servidor que mantém muitas conexões, por exemplo, com múltiplos clientes, essa sobrecarga pode ser muito grande. Há, então, uma necessidade de substituir uma matriz compartilhada sem incorrer em uma sobrecarga grande.

Referências

[1] “Device and method sharing a matrix for use in a cryptographic protocol”, depósito europeu op 6 de março de 2017, número do pedido 17159296.7. Documento do Procurador 2017P01555

[2] Ludo Tolhuizen, Ronald Rietman and Oscar Garcia-Morchon, “Improved key reconciliation method”, Cryptology ePrint Archive, Report 2017/295, https://eprint.iacr.org/2017/295

[3] (Frodo): J. Bos et al, “Frodo: Take off the ring! Practical, Quantum-Secure Key Exchange from LWE”, Cryptology ePrint Archive, Report 2016/659, https://eprint.iacr.org/2016/659

[4] (New Hope): E. Alkim et al, “Post-quantum key exchange – a new hope”, Cryptology ePrint Archive, Report 2015/192, https://eprint.iacr.org/2015/1092

[5] (New Hope Simple): E. Alim et al, “NewHope without reconciliation”, Cryptology ePrint Archive, Report 2016/1157, https://eprint.iacr.org/2016/1157

[6] (Kyber): J. Bos et al, “CRYSTALS -- Kyber: a CCA-secure module-lattice-based KEM”, Cryptology ePrint Archive, Report 2017/634, https://eprint.iacr.org/2017/634

[7] (spKEX): S. Bhattacharya et al, “spKEX: an optimized lattice-based key exchange”, Cryptology ePrint Archive, Report 2017/709, https://eprint.iacr.org/2017/709

Cada uma das referências [1] a [7] está incluída no presente documento a título de referência.

Sumário da invenção

[014] É proposto um nó de rede que implementa uma criptográfica que depende da complexidade de um problema de reticulado e que emprega uma matriz compartilhada. Conforme destacado na seção de antecedentes da invenção, existem muitas tais operações criptográficas. Os inventores perceberam que a sobrecarga pode ser reduzida introduzindo-se um grupo compartilhado. Em vez de gerar a matriz compartilhada diretamente, o primeiro e segundo nó compartilham primeiramente

um grupo compartilhado. O grupo compreende múltiplos números inteiros ou polinômios, assim como faz a matriz compartilhada, ou o grupo compreende múltiplos coeficientes que podem ser usados para os polinômios. Em uma modalidade, o grupo compartilhado não é alterado entre os dois nós, e permanece igual, ou pelo menos permanece igual ao longo de múltiplas execuções da operação criptográfica. No entanto, para cada operação criptográfica, uma nova matriz compartilhada é construída a partir do grupo compartilhado. Isso é feito selecionando-se funções que mapeiam elementos do grupo compartilhado para entradas na matriz compartilhada ou que mapeiam coeficientes do grupo compartilhado para o coeficiente da matriz compartilhada. As funções de seleção dependem do índice no grupo compartilhado ou do índice na matriz compartilhada, mas curiosamente também dependem de um parâmetro de seleção. Um parâmetro de seleção muda a maneira que as entradas da matriz compartilhada são selecionadas a partir do grupo compartilhado. Os inventores perceberam que as funções de seleção variável exigem sobrecarga muito pequena e, ademais, tal função pode ser fácil de computar. Por exemplo, uma função de seleção pode ser uma permutação que depende do parâmetro de seleção que permuta o grupo compartilhado ou uma parte dele e o atribui a um subgrupo da matriz ou aos coeficientes. A matriz A pode ser particionada em um número de tais subgrupos. Cada subgrupo pode ser obtido do mesmo grupo compartilhado ou cada subgrupo pode ser obtido de uma parte do grupo compartilhado. Ainda assim, as partes do grupo compartilhado que são mapeadas para os subconjuntos da matriz compartilhada podem se sobrepor, até mesmo de maneira significativa.

[015] Como resultado, o grupo compartilhado precisa conter um número significativamente menor de dados que

a matriz compartilhada. Isso significa que menos dados precisam ser mantidos no armazenamento, mas também que a construção da matriz compartilhada pode ser significativamente mais rápida. Por exemplo, considere-se como uma comparação uma situação em que a matriz compartilhada é gerada a partir de uma semente por uma função (pseudo)aleatória. Isso exige um número relativamente maior de operações de função aleatória. No entanto, mediante a geração primeiramente de um grupo compartilhado, o que também pode ser feito a partir de uma semente, menos aplicações de função aleatória são necessárias. Visto que parte da aplicação de função de número aleatório é substituída pela função de seleção, que pode ser, por exemplo, implementada com permutações que exigem menos recursos, o tempo para gerar a matriz compartilhada é reduzido. Isso tem, também, uma vantagem em operações criptográficas que efetivamente usam uma matriz compartilhada, mas não exigem que ela seja frequentemente substituída.

[016] Conforme destacado acima, há várias operações criptográficas que usam uma matriz compartilhada. Algumas delas usam polinômios e algumas usam números inteiros. Os inventores perceberam que é possível descrever muitos desses esquemas de uma maneira que parametrize grandes classes de criptografia baseada em reticulados. Por exemplo, tal descrição pode usar apenas alguns parâmetros de entrada, que mudam a natureza da criptografia subjacente: um parâmetro de dificuldade (d), e um parâmetro de estrutura (n) que juntos podem ser suficientes para especificar o tipo de criptografia que é utilizada. Tanto o parâmetro de dificuldade (d) quanto um parâmetro de estrutura (n) são números inteiros positivos. Mediante o aumento do parâmetro de dificuldade, a estrutura

do problema permanece a mesma, porém maior, e, consequentemente, instanciações mais difíceis são usadas. Um parâmetro de dificuldade é por vezes chamado de um parâmetro de segurança. Em uma modalidade, o parâmetro de dificuldade está relacionado à dimensão do reticulado subjacente. Um valor maior do parâmetro dificuldade leva a uma dimensão superior do reticulado subjacente e, desse modo, a um problema mais difícil para um invasor resolver. O parâmetro de estrutura está relacionado à quantidade de estrutura algébrica no sistema. Um valor mais alto de parâmetro de estrutura implica mais estrutura algébrica. Conjectura-se que os valores mais altos do parâmetro de estrutura podem ser mais fáceis de invadir, visto que o reticulado tem mais regularidade. Por exemplo, caso a pesquisa avançada constate um modo na progressivamente melhor de solucionar problemas de reticulados, então, um usuário pode responder aumentando-se o parâmetro de dificuldade. Por outro lado, caso a pesquisa avançada constate um modo radialmente melhor de solucionar um tipo particular de problema de reticulados, então, o usuário pode responder com a diminuição do parâmetro de dificuldade.

[017] Por exemplo, um dispositivo criptográfico, conforme definido nas reivindicações, pode receber ainda o parâmetro de dificuldade (d) e o parâmetro de estrutura (n) que juntos são suficientes para especificar o tipo de criptografia que é usada. Por exemplo, é definido em uma modalidade um esquema que pode eficientemente representar múltiplos problemas subjacentes, a saber, RLWE, RLWR, módulo de RLWE, módulo de RLWR e LWE e LWR. Isso claramente não é necessário, no entanto, e uma modalidade pode selecionar uma dimensão fixa da matriz, por exemplo, $k \times k$, e se forem

utilizados polinômios, a modalidade pode selecionar um polinômio de redução de grau *n*

[018] O dispositivo criptográfico é um dispositivo eletrônico. Por exemplo, o dispositivo criptográfico pode ser um dispositivo eletrônico móvel, por exemplo, um telefone móvel. Por exemplo, o dispositivo criptográfico pode ser um decodificador, um cartão inteligente, um computador etc. O método para realizar uma operação criptográfica descrito no presente documento pode ser aplicado em uma gama de aplicações práticas. Tais aplicações práticas incluem aplicações financeiras, comunicações confidencial ou autenticada etc.

[019] Um método de acordo com a invenção pode ser implementado em um computador como um método implementado por computador, ou em um hardware dedicado, ou em uma combinação de ambos. O código executável para um método de acordo com a invenção pode ser armazenado em um produto de programa de computador. Exemplos de produtos de programa de computador incluem dispositivos de memória, dispositivos de armazenamento óptico, circuitos integrados, servidores, softwares online, etc. De preferência, o produto de programa de computador contém código de programa não transitório armazenado em uma mídia legível por computador para realizar um método de acordo com a invenção, quando o dito produto de programa é executado em um computador.

[020] Em uma modalidade preferencial, o programa de computador compreende código de programa de computador adaptado para executar todas as etapas de um método de acordo com a invenção, quando o programa de computador for executado em um computador. De preferência, o programa de computador é incorporado em uma mídia legível por computador.

[021] Um outro aspecto da invenção apresenta um método para produção do programa de computador disponível para transferência via download. Este aspecto é usado quando o programa de computador é transferido via upload para, por exemplo, a App Store da Apple, a Play Store da Google ou a Windows Store da Microsoft, e quando o programa de computador está disponível para download a partir de tal loja.

[022] Breve descrição dos desenhos

[023] Detalhes adicionais, aspectos e modalidades da invenção serão descritos, apenas a título de exemplo, com referência aos desenhos. Os elementos nas figuras são ilustrados por uma questão de simplicidade e clareza, e não estão necessariamente em escala. Nas figuras, os elementos que correspondem a elementos já descritos podem ter as mesmas referências numéricas. Nos desenhos,

[024] a Figura 1 mostra esquematicamente um exemplo de uma modalidade de uma rede de concordância de chaves,

[025] a Figura 2 mostra esquematicamente um exemplo de uma modalidade de um método de troca de chave eletrônica,

[026] a Figura 3a mostra esquematicamente uma mídia legível por computador que tem uma parte gravável que compreende um programa de computador de acordo com uma modalidade,

[027] a Figura 3b mostra esquematicamente uma representação de um sistema processador de acordo com uma modalidade,

[028] a Figura 4 mostra esquematicamente um exemplo de uma modalidade de uma matriz compartilhada.

Lista de números de referência nas figuras 1 a 2:

| | |
|---|---|
| 100 | uma rede de concordância de chaves |
| 110 | um nó de rede do tipo iniciador |
| 120 | uma interface de comunicação |
| 130 | uma unidade de matriz compartilhada |
| 131 | um grupo compartilhado |
| 140 | uma unidade de matriz de chave privada |
| 150 | uma unidade de matriz de chave pública |
| 160 | uma unidade de chave compartilhada |
| 162 | uma chave bruta |
| 164 | dados de reconciliação (*h*) |
| 166 | uma chave compartilhada |
| | |
| 210 | um nó de rede do tipo respondente |
| 220 | uma interface de comunicação |
| 230 | uma unidade de matriz compartilhada |
| 240 | uma unidade de matriz de chave privada |
| 250 | uma unidade de matriz de chave pública |
| 260 | uma unidade de chave compartilhada |
| 262 | uma chave bruta |
| 264 | dados de reconciliação (*h*) |
| 266 | uma chave compartilhada |
| | |
| 1000 | mídia legível por computador |
| 1010 | uma parte gravável |
| 1020 | um programa de computador |
| 1110 | um circuito integrado (ou circuitos integrados) |
| 1120 | uma unidade de processamento |
| 1122 | uma memória |
| 1124 | um circuito integrado dedicado |
| 1126 | um elemento de comunicação |

1130 uma interconexão

1140 um sistema processador

Descrição detalhada das modalidades

[029] Embora a invenção seja suscetível de realização de várias formas diferentes, são mostradas nos desenhos, e serão aqui descritas em detalhe, uma ou mais modalidades específicas, com o entendimento de que a presente revelação deve ser considerada como exemplificadora dos princípios da invenção e não se destinam a limitar a invenção às modalidades específicas ilustradas e descritas. Na descrição a seguir, para melhor entendimento, os elementos das modalidades são descritos em funcionamento. Entretanto, ficará evidente que os respectivos elementos são dispostos de modo a executar as funções sendo descritas, conforme executadas pelos mesmos. Adicionalmente, a invenção não se limita às modalidades, e a invenção se encontra em todo e qualquer recurso inovador ou em toda e qualquer combinação de recursos descritos acima ou mencionados em reivindicações dependentes mutuamente diferentes.

[030] Vale lembrar que há diversos esquemas baseados em reticulados para o modelo de um protocolo de troca de chaves (KEX), um método de encapsulação de chave (KEM), criptografia de chave pública (PKE), assinaturas digitais etc. Os exemplos desses esquemas são:

[031] Frodo, que é uma KEX baseada no problema de Aprendizagem com erros (LWE).

[032] NewHope que é uma KEX baseada no problema Aprendizagem em anel com erros (RLWE).

[033] NewHopeSimple que uma KEM projetada para ataque de texto simples escolhido (CPA) com base em RLWE.

[034] Kyber que é uma CPA KEM e CCA KEM com base no problema de LWE de módulo.

[035] spKEX que é uma KEX baseada no problema de Aprendizagem com arredondamento (LWR).

[036] Cada um dos esquemas acima implementa um protocolo criptográfico (por exemplo, uma KEX, uma KEM, PKE, assinatura digital) que é baseado em um único problema de reticulados: ou LWE ou RLWE ou LWE de módulo para um anel fixo ou LWR.

[037] NewHope e NewHopeSimple se baseiam apenas em RLWE usando de anel $\frac{Z_q[x]}{x^{1024}+1}$

[038] Kyber se apenas na combinação de k módulos, sendo que cada módulo são polinômios em $\frac{Z_q[x]}{x^{256}+1}$

[039] A spKEX se baseia apenas em LWR

[040] Frodo se baseia apenas em LWE

[041] Embora todos os problemas acima estejam relacionados, alguns deles são considerados mais arriscados devido ao fato de que não se baseiam em estruturas especiais. Por outro lado, o uso de uma estrutura em anel permite mensagens menores. Embora atualmente os presentes ataques que empregam tais estruturas sejam desconhecidos, tais ataques podem ser conhecidos futuramente. Todos os esquemas baseados nesses problemas específicos (por exemplo, RLWE ou LWE de módulo) seriam, então, corrompidos. Desse modo, há uma necessidade de uma posição de fallback. Desse modo, vários esquemas precisam ser implementados e testados, alguns baseados em problemas mais arriscados, porém com melhores propriedades de desempenho e alguns baseados em problemas bem entendidos, porém com um desempenho menos satisfatórios.

[042] Os inventores perceberam que os problemas de reticulados estão relacionados e constataram um esquema que pode instanciar todos esses problemas, incluindo: RLWE, RLWR, LWE de módulo, LWR de módulo e LWE e LWR. A seguir são descritos alguns exemplos de um criptográfico no qual diferentes reticulados são criados com base em um parâmetro de dificuldade $d$, e um parâmetro de estrutura $n$.

[043] Mediante a alteração de parâmetros nesse esquema, é possível, portanto, instanciar os protocolos (KEX, KEM, PKE etc.) baseado em problemas subjacentes e que exibem diferentes propriedades de desempenho.

[044] Esse esquema pode ser aplicado a múltiplos aplicações com diferentes demandas de segurança/desempenho. Por exemplo, os documentos altamente secretos podem ser descriptografados com o uso das configurações do esquema para LWE ao passo que a concordância de chaves em uma comunicação sem fio de recurso limitado pode ser baseada em parâmetros de LWR de Anel. O esquema tem vantagens adicionais: em primeiro lugar, ele reduz o tamanho do código de modo que seja necessário menos memória. Em segundo lugar, o esforço para revisão de código é minimizado visto que o algoritmo precisa ser revisado e verificado. Por fim, tal esquema se prepara para um cenário de transição potencial no qual se basear em um problema “mais fraco” (por exemplo, RLWE) não é mais seguro, e são necessários novos algoritmos que se baseiam em uma problema “mais forte” (por exemplo, com base em LWE). O esquema não se baseia em duas especificações diferentes, uma para o Problema 1 e outra para o Problema 2 etc., mas sim no fato de que o mesmo algoritmo é usado para instanciar diferentes problemas, sendo a única diferença os parâmetros de entrada.

[045] A presente invenção usa um problema de reticulados de dimensão d - em que d é um parâmetro de entrada. A dimensão de reticulados pode ser aquela que corresponde a RLWE, RLWR, LWE de Módulo, LWR de Módulo, LWE, LWR. O esquema se baseia em uma matriz que contém $k\,x\,k$ entradas, sendo que cada entrada é um elemento em Zq[x]/f[x], em que f[x] é um polinômio de redução de grau n, e n é um parâmetro de entrada. k é definido como sendo d/n e não é um parâmetro de entrada; em modalidades, presume-se que d é um múltiplo de n.

[046] Desse modo, dado uma dimensão fixa de reticulados d e o grau do polinômio de redução n, é possível obter o número de entradas ($k^2$) da matriz. Note-se que esta é uma diferença fundamental com reticulados modulares. Por exemplo, em tais reticulados, alguns autores otimizam as operações de anel em um anel de polinômio (a saber, $f(x) = x^{256} + 1$) e usam o número de módulos para "aumentar" a dimensão da reticulados como d=256*k. Esse esquema pode ser instanciado apenas como RLWE de módulo ou RLWE que é pequeno demais para fins de segurança.

[047] Se n=1, então, isso representa uma matriz para LWE ou LWR.

[048] Se n=d, então, isso representa uma matriz para RLWE ou RLWR.

[049] Se 1<n<d, então, isso representa uma matriz para o LWE de módulo ou LWR de módulo.

[050] Sem perder generalidade, é possível observar que nas seções a seguir, atenção foi dada em maior parte apenas para dois casos, a saber, n=1 e n=d.

KEX baseada em LWR e em LWR EM ANEL DE NTRU

[051] A seguir, o ANEL de NTRU será usado para ilustrar as modalidades da presente invenção. Na prática, outros anéis podem ser usados também como polinômios ciclotômicos (primo) que consideram q como um primo que garante que n=1 (mod p). Alternativamente, é possível obter também polinômios ciclotômicos (primo) com q sendo uma potência de dois. Uma restrição adicional é que o polinômio ciclotômico (primo) é módulo 2 irredutível.

[052] O anel de NTRU é $f(x) = x^n - 1$ em que *n* é um número primo. Então, para um dado problema de dimensão de reticulados d, é possível instanciar o sistema – por exemplo – com $n = d$ ou $n = 1$. Se $n = d$, então, temos uma LWR EM ANEL DE NTRU e se $n = 1$, então, há um esquema de LWR. É possível também obter um parâmetro de entrada d>n e d sendo um múltiplo do número primo n de modo que se obtenha um LWR de Módulo com o uso do anel de NTRU.

[053] Tabela 1 - Descrição de protocolo KEX de alto nível. Note-se que round(vector, p, q) indica a realização de arredondamento com o uso dos módulos p e q. Note-se os detalhes sobre como a reconciliação de chave é feita são ignorados, visto que eles não são fundamentais para a presente descrição. As informações adicionais abaixo contêm os protocolos PKE, CPA-KEM e CCA-KEM, incluindo todos os detalhes necessários.

| Iniciador | Respondente |
|---|---|
| Entrada n e d em que n é primo e n é um divisor de d<br>Criar matriz A com d/n x d/n entradas em Z [x]/f[x]<br>Criar segredos s que contêm d/n entradas em Z[x]/f(x) | |

| | |
|---|---|
| Criar chave pública b=round (A s, p, q) com os elementos d/n em Zp[x]/f(x), em que A s é o produto da matriz A e do vetor de secreto s, do módulo computado f(x) e do módulo q.<br>Enviar (b, A) | |
| | Criar r secreto que contém elementos d/n em Z[x]/f(x).<br>Criar chave pública u=round (r^t A, p, q) com d/n entradas em Zp[x]/f(x), em que r^t A é o produto de matriz do vetor secreto transposto r e da matriz A, módulo computado f(x) e módulo q<br>Computar a chave bruta rkr=(r^t b) (mod p) que contém d/n entradas em Zp[x]/f(x), em que r^t é o produto de matriz do vetor secreto transposto r e da matriz b, computados módulo f(x) e módulo p.<br>Computar dados de reconciliação (h) a partir da rkr<br>Enviar (u, h) |
| Computar a chave bruta rki=u s (mod p) que contém os elementos d/n em Zp[x]/f(x), em que u s é o produto da chave pública u e do vetor secreto s, do módulo computado f(x) e do módulo p<br>Computar a chave final de h e rki. | |

[054] A partir do supracitado, é possível observar que dependendo da escolha de n, o problema subjacente é RLWR (se n=d) ou LWR (n=1). Em ambos os casos, o problema do reticulado subjacente tem dimensão d. Note-se que a técnica teria duas implementações diferentes do algoritmo, a saber:

[055] Tabela 2 - Note-se que o arredondamento necessário é ignorado quando as chaves públicas b e u são computadas, visto ele não é necessário para a descrição. Os detalhes sobre como a reconciliação de chave de ser realizada também são ignorados, consultar as referências.

KEX baseada em LWR (nota que algumas operações estão faltando para ilustrar o protocolo)

| Iniciador | Respondente |
|---|---|
| Criar A com elementos d x d em Z<br>Criar s secreto, um vetor de número inteiro de comprimento d (Z^d)<br>Criar chave pública<br>Valores b=round (As, p, q) em Z_p^d em que A s é o produto da matriz A e do vetor s, | |

KEX baseada em RLWR (nota que algumas operações estão faltando para ilustrar o protocolo)

| Iniciador | Respondente |
|---|---|
| Criar a em Zq[x]/f(x)<br>Criar s secreto em Z[x]/f(x)<br>Criar chave pública<br>b=round (a*s, p, q) em Zp[x]/f(x) em que * é multiplicação de polinômios, computados módulo f e módulo q.<br>Enviar (b, a) | |

| | |
|---|---|
| do módulo q computado<br>Enviar (b, A) | |
| | Criar r secreto em Z^d<br>Criar chave pública u=round (r^t A, p, q) em Z_p^d em que r^t A é o módulo q computado.<br>Computar a chave bruta rkr=(r ^ t b) (mod p) em Z_p), em que r^t b é o produto de matriz do vetor secreto transposto r e da matriz b, computado módulo p.<br>Computar dados de reconciliação (h) a partir da rkr<br>Enviar (u, h) |
| Computar chave bruta rki=(u s) (mod p)<br>Computar chave final de h e rki, | |

| | |
|---|---|
| | Criar r secreto em Z[x]/f(x)<br>Criar chave pública de u=round (a*r, p, q) em Zp[x]/f(x) em que a*r é computado módulo f e módulo q.<br>Computar chave bruta rki=b*r em Zp[x]/f(x) em que * é multiplicação de polinômios, computados módulo f e módulo p.<br>Computar dados de reconciliação (h) a partir da rkr<br>Enviar (u, h) |
| Computar chave bruta rki=u*r em Zp[x]/f(x) em que * significa multiplicação de polinômios, computados módulo f e módulo p. | |

| | | | |
|---|---|---|---|
| em que u s é o produto da chave pública u e vetor secreto s, módulo p computado. | | Computar a chave final de h e rki. | |

Note-se que: rki e rkr são as chaves brutas do iniciador e do respondente, respectivamente.

[056] Conforme descrito nos antecedentes e declaração do problema, os esquemas atuais na Tabela 2 se baseiam em um problema único e, desse modo, são otimizados com parâmetros e algoritmos levemente diferentes. Por exemplo, os problemas RLWE se baseiam geralmente no anel Zq[x]/x^n+1 em que n é uma potência de dois e q é um número primo de modo que a Transformada Teórica de Número (NTT) também possa ser usada. Essas escolhas não são ideais para combinação com versões não anel visto que é necessário que q seja um primo que torna as operações modulares mais complexas durante a realização das operações de matriz.

[057] O esquema na Tabela 1 é definido para funcionar com parâmetros e rotinas similares. Esse é um motivo particular para usar o anel de NTRU nessa descrição, visto que ele se baseia em um q que é uma potência de dois que também é uma boa opção para esquemas de LWE/LWR. Isso nos permite também usar a mesma rotina para a etapa de reconciliação de chave que se baseia no fato de que tanto q quanto i são potências de dois [2]. Esses atributos são explicados detalhadamente nas informações adicionais a seguir, explicando a modalidade para CPA PKE, CPA KEM e CCA KEM.

[058] Além disso, deve-se notar que no protocolo explicado na Tabela 1, há referências a round() que é uma função

que realiza o arredondamento, conforme definido no problema de LWR.

Implementação de referência (parte 1)

[059] Uma implementação de referência do esquema na Tabela 1 tem as seguintes rotinas para obter os elementos de chave pública (b e u) e chaves brutas (rki e rkr):

Computação de chaves públicas:

```
Result[] = Computação de chave pública (A[,],s[])
              Result[] = 0
For(i=0 to d/n)
                    For(j=0 to d/n)
                         Result[i] =
Add_elements(Result[i], Multiply_elements[A[i,j],s[j])
       C[]=Add_elements(A[],B[])
              For(i=0 to n)
                    C[i]=(A[i]+B[i] (mod q))
       C[]=Multiply_elements(A[],B[])
              C[] =0
                    For(i=0 to n)
                        For(j=0 to n)
                          C[i]=C[i]+ A((i-j)mod n)*B[j] mod q
```

[060] Note-se que nas rotinas acima, ã etapa de arredondamento que usa os módulos p e q, conforme a Tabela 1, não está incluída. No entanto, isso não é essencial para a presente invenção. Caso estivesse sendo feita uma versão de LWE desse esquema, o arredondamento deve ser substituído pela adição de ruído (por exemplo, ruído Gaussiano da média 0 e pequeno desvio padrão) em Z_q[x]/f(x).

Computação de chaves brutas:

```
Result[] = Computação de chave bruta (b[],s[])
```

```
                Result[] = 0
For(i=0 to d/n)
                        Result[i]  =  Add_elements(Result[i],
Multiply_elements[b[i],s[i])
    C[]=Add_elements(A[],B[])
        For(i=0 to n)
          C[i]=(A[i]+B[i] (mod p) (mod x^n-1)
    C[]=Multiply_elements(A[],B[])
        For(i=0 to n)
          C[i]=(A[(n-i)mod(n)]+B[i]) (mod p) (mod x^n-1)
```

[061] Note-se que nos algoritmos de computação de chave bruta, mod p é escrito, uma vez que a Aprendizagem com arredondamento é considerada. Alternativamente, pode-se multiplicar primeiramente cada coeficiente de A e B com q/p, arredondar para o número inteiro mais próximo, realizar adição ou multiplicação módulo q e escalar o resultado final por multiplicação em termos de coeficiente com p/q, seguido de arredondamento para o próximo número inteiro. Isso pode ter vantagens de segurança. Para uma versão Aprendizagem com erros (com anel) da presente invenção, é adicionado ruído em vez de da realização de arredondamento, e as computações são o módulo q.

Implementação de referência (parte 2)

[062] Constata-se que os segredos associado ao iniciador e ao respondente em um protocolo (KEX, KEM, PKE, etc.) pode ser mais que um único vetor que contém elementos d/n em Zq[x]/f(x), porém podem conter n_bar e m_bar desses vetores. É preferencial que n_bar e m_bar sejam maiores que um para gerar bits de chave suficientes. Desse modo, todas as operações no esquema em uma implementação de referência podem

ser representadas como a multiplicação de duas matrizes cujos elementos estão em Zq[x]/f(x).

[063] Para isso, é necessário a seguinte função genérica:

[064] C = Multiplicar (A,B,A_c, A_r, B_c, B_r, n)

[065] em que A e B são as matrizes de entrada das dimensões (A_c x A_r) e (B_c x B_r) respectivamente. Cada entrada na matriz contém um elemento em Zq[x]/f(x) que pode ser representado como n elementos em Zq. A matriz de saída C tem a dimensão (C_c x C_r) = (A_r x B_c).

[066] Além disso, em alguns casos, é necessária uma função para transpor as matrizes. Para isso, é necessária uma função:

[067] B = Transpor (A, A_c, A_r, n)

[068] em que A é a matriz de entrada de dimensões (A_c x A_r). Cada entrada na matriz contém um elemento em Zq[x]/f(x) que pode ser representado como elementos em Zq. A matriz de saída B tem a dimensão (B_c x B_r) = (A_r x A_c).

[069] Um modo particular de codificar as informações em uma matriz é usar um vetor que armazena nos elementos de nível mais baixo em Zq. Por exemplo, se =2^14, então, dois bytes podem ser usados para armazenar cada elemento. Uma matriz A de dimensão (A_r x A_c) com elementos em Zq[x]/f(x) pode ser armazenada em um vetor de comprimento A_r*A_c*n*2 bytes (presumindo que q<=2^16). Esse vetor pode codificar A: (i) linha por linha, (ii) em cada linha coluna por coluna e (iii), então, os n elementos em Zq.

[070] As duas funções acima (Multiplicar() e Transpor()) podem acessar o vetor usado para armazenar as

matrizes da maneira adequada. Esse também é o motivo pelo qual n é passado como um parâmetro de entrada para essas duas funções.

Implementação otimizada da computação de chave pública (parte 1):

[071] Para um parâmetro de segurança fixo d, a implementação de referência acima é rápida para o caso n=d, visto que a matriz A contém uma única entrada que é um polinômio em $Zq[x]/x^n - 1$, presumindo que n_bar=m_bar=1. Para $n = 1$, uma implementação de referência será lenta visto que A contém $d^2$ entradas, sendo que cada entrada é um elemento em Zq e a multiplicação escalar é realizada como uma multiplicação de polinômio genérico.

[072] Para uma implementação ideal, pode-se explorar o fato de que no caso | n=d, as operações de multiplicação de polinômios podem ser expressadas como a multiplicação de uma matriz dxd=nxn sobre Zq e um vetor de comprimento d com entradas de Zq:

$$a * s = \begin{pmatrix} a_0 & a_{n-1} & \dots & a_1 \\ a_1 & a_0 & \dots & a_2 \\ \dots & \dots & \dots & \dots \\ a_{n-1} & a_{n-2} & \dots & a_0 \end{pmatrix} \begin{pmatrix} s_0 \\ s_1 \\ \dots \\ s_{n-1} \end{pmatrix}$$

[073] Usando-se esse fato, durante a implementação da computação de chave pública, pode-se sempre presumir uma matriz A de entrada que conterá $d^2$ elementos diferentes para n=1 (LWR) e d elementos diferentes para n=d (RLWR). A única diferença entre os algoritmos durante as multiplicações de matriz é o fato de que a multiplicação de matrizes para o respondente é feita a partir da esquerda para n=1 e a partir da direita para n=d.

[074] Alternativamente, os coeficientes de polinômios podem ser organizados como vetores de linha, e o seguinte é usado:

$$a * s = (s_0, s_1, \dots, s_{n-1}) \begin{pmatrix} a_0 & a_1 & \dots & a_{n-1} \\ a_{n-1} & a_0 & \dots & a_{n-2} \\ \dots & \dots & \dots & \dots \\ a_1 & a_2 & \dots & a_0 \end{pmatrix}$$

Implementação otimizada da computação de chave pública (parte 2)

[075] Conforme apresentado em um depósito Philips relacionado [1], A pode ser renovado, quando recebe uma matriz mestre A_master, aplicando-se uma permutação. A permutação pode ser, por exemplo, um deslocamento cíclico (por um deslocamento aleatório (entre 0 e n-1) de um número n' (0 <= n' < n) das linhas em A_master.

[076] Uma extensão natural disso é considerar A_master como um vetor a_master de comprimento L e aplicar uma permutação sobre ele para obter as linhas de A. Três casos são diferenciados com permutações muito específicas.

[077] Caso 1: $L = d^2$ e a permutação para obter a linha $i$ de $A$ compreende obter os d elementos consecutivos começando na posição $i * d$ de $a_{master}$ e girando-os ciclicamente em um deslocamento aleatório entre 0 e d-1.

[078] Caso 2: $L \approx q$ e a permutação para obter a linha $i$ de $A$ compreende obter os $d$ elementos consecutivos, começando no elemento $e_i$ de $a_{master}$ em que $e_i$ é um valor escolhido aleatoriamente em $[0, L-1]$ e que termina no elemento $e_i + d \ (mod\ L)$ (desse modo, considerando uma estrutura cíclica).

[079] Caso 3: $L = d$ e a permutação para obter a linha $i$ de $A$ compreende tomar os $d$ elementos consecutivos

(considerando também uma estrutura cíclica) que começa no elemento $i$ de $a_{master}$.

[080] Obviamente, os três casos acima podem ser generalizados com o uso de outros tipos de permutações ou fazendo $L > d^2$. Para os três casos nessa abordagem:

[081] O Caso 1 implementa o esquema descrito no presente documento para $n = 1$. Caso $a_{master}$seja computada aleatoriamente, então, o $A$ resultante não pode ser diferenciado a partir de uma matriz aleatória. Visto que os elementos são girados por um deslocamento aleatório, em seguida, recebem um a_master fixo, essa abordagem pode obter eficientemente um A novo.

[082] O Caso 3 implementa o esquema para $n = d$ e fornece um modo eficiente para obter a representação de matriz de $a$ na convolução de polinômios $a * s$.

[083] O Caso 2 é um caso entre uma estrutura de anel e uma estrutura de não anel. Para esse caso, não é possível ter uma redução para (LWR), visto que haverá alguma sobreposição entre as linhas em A visto que L<d^2. Desse modo, é possível distinguir a matriz resultante $A$ de uma matriz aleatória. Entretanto, de um ponto de vista prático, essa abordagem destrói a estrutura de anel único no $A$ resultante, visto que ela contém muito mais elementos. De maneira eficaz, cada linha usa um anel diferente.

[084] Caso a abordagem acima seja incorporada à implementação otimizada (Parte 1), pode-se implementar a multiplicação de matrizes em termos de $a_{master}$ de um determinado comprimento $L$, uma permutação de alguns recursos específicos e o secreto $s$. Dependendo da escolha do comprimento $L$ e da

permutação, o mesmo algoritmo implementa eficientemente a multiplicação de matrizes para ambos $n = 1$ e $n = d$.

[085] Note-se que no caso 1, a_master pode ser novamente usada, ela e matrizes novas A são obtidas variando-se as permutações de linha. No caso 3, é aconselhável para atualizar regularmente a_master.

[086] Isso pode ser resumido na tabela a seguir em que é observado que, dependendo da escolha de L, é possível obter diferentes desempenhos e garantias de segurança. Note-se que a obtenção de L=q não tem benefícios do ponto de vista de largura de banda, uma vez que a estrutura é destruída e mais informações são transmitidas para transmitir A; entretanto, em termos de CPU, a abordagem é mais eficiente visto que menos números aleatórios precisam ser computados.

| Valor de L | Corresponde a | O problema é reduzido para | Desempenho (largura de banda) | Desempenho (CPU) |
|---|---|---|---|---|
| d | n=d | RLWR | A partir de RLWR | A partir de RLWR |
| q | n=1 | RLWR | A partir de LWR | Perto de RLWR |
| d^2 | n=1 | LWR | A partir de LWR | A partir de LWR |

Implementação otimizada (parte 3)

[087] A mesma ideia de permutação pode ser aplicada à computação da chave bruta sabendo que em ambos os casos a chave pública contém *d* elementos em Zp. Para $n = 1$, aqueles são, na realidade, *d* elementos em $Zp = Zp[x]/x - 1$. Para $n = d$, aqueles *d* elementos são os coeficientes de um polinômio em $Zp[x]/x^d - 1$.

[088] Para $n = 1$, os $d$ elementos da chave pública recebida são processados na ordem recebida, porém para o caso $n = d$, os d elementos precisam, primeiramente, ser redispostos para se assemelharem à ordem da primeira fileira da matriz que implementa a avaliação de polinômios. Desse modo, pode=se definir duas transformações:

| | |
|---|---|
| $T_{n=1} = \begin{pmatrix} 1 & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & 1 \end{pmatrix}$ | $T_{n=d} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 1 \\ \dots & \dots & \dots & \dots \\ 0 & 1 & \dots & 0 \end{pmatrix}$ |

[089] Além disso, consideramos que dois permutações são aplicadas para obter uma matriz que se assemelha às multiplicações de polinômios

| | |
|---|---|
| $P_{n=1} = \begin{pmatrix} 1 & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & 1 \end{pmatrix}$ | $P_{n=d} = \begin{pmatrix} 0 & 0 & \dots & 1 \\ \dots & \dots & \dots & \dots \\ 0 & 1 & \dots & 0 \\ 1 & 0 & \dots & 0 \end{pmatrix}$ |

[090] Desse modo, um algoritmo otimizado para obter a chave bruta funciona de acordo com o seguinte:

[091] Result[] = Computação de chave bruta (b[],s[])

```
Result[] = 0
For(i=0 to n)
   Para j=0 para d
      Result[i] = Result[i] + T_n(b)[(P_n(j)+i)mod(d)] * s(j)
```

[092] Note-se que $T_d$ é exigido, uma vez que o vetor de coeficiente $b = (b_0, b_1, \dots, b_{n-1})^T$ é recebido e é necessário obter a linha superior da matriz circulante com b como a coluna mais à esquerda.

[093] No algoritmo acima, $T_n$ atua sobre $b$ para redispor os elementos adequadamente e $P_n$ realiza um deslocamento em $T_n$(b) a fim de realizar as operações de redução de módulo.

Implementação otimizada (parte 4)

[094] Um aspecto importante para a implementação otimizada é a realização da permutação em a_master. Isso exige a execução de reduções módulo L e módulo d. As desvantagens são em relação ao desempenho e aos ataques de canal lateral. Para evitar essas reduções modulares, por exemplo, em um vetor **v**:

$$\boldsymbol{v}[(i + P[j])(mod\ d)]$$

a implementação otimizada pode trabalhar com um vetor **v_star** que contém apenas uma cópia de vetor.

$$\boldsymbol{v_star} = \boldsymbol{v} \mid \boldsymbol{v}$$

[095] Em que | representa concatenação. Isso aumenta as exigências de memória, porém evitar a redução modular.

[096] Constata-se também que essa abordagem pode possibilitar a implementação de outros anéis de maneira eficiente. Por exemplo, caso seja usado o anel Zq[x]/x^n+1, então, v_star será obtido como:

$$\boldsymbol{v_star} = \boldsymbol{v} \mid -\boldsymbol{v}$$

De fato, teria-se, então,

$$a * s = \begin{pmatrix} a_0 & -a_{n-1} & \cdots & -a_1 \\ a_1 & a_0 & \cdots & -a_2 \\ \cdots & \cdots & \cdots & \cdots \\ a_{n-1} & a_{n-2} & \cdots & a_0 \end{pmatrix} \begin{pmatrix} s_0 \\ s_1 \\ \cdots \\ s_{n-1} \end{pmatrix}$$

[097] Logo, A tem sinais de menos acima da diagonal; na realidade, $A(i,j) = a_{i-j}$ se $i \geq j$ e $A(i,j) = -a_{n+i-j}$ se $i < j$. Visto que $a_0 = A(0,0)$ e $a_k = -A(0,n-k)$ para $1 \leq k \leq n-1$, conclui-se que $A(i,j) = -A(0,n-i+j)$ se $i > j$ e $A(i,j) = A(0,j-i)$ se

$j \geq i$. Logo, $A(i,j) = v(n-i+j)$ onde $v(k) = -A(0,k-n)$ se $k < n$ e $v(k) = A(0,k-n)$ para $k \geq n$.

CPA-PKE, CPA-KEM e CCA-KEM baseada em LWR e LWR em ANEL DE NTRU

[098] As informações adicionais abaixo descrevem como construir protocolos KEM CPA PKE, CPA KEM e CCA com base nas ideias descritas até agora. Esses protocolos podem ser, então, instanciados para se basear em LWR e LWR em ANEL de NTRU apenas com o uso de diferentes parâmetros de configuração.

[099] O anel de NTRU tem polinômio de redução $f(x) = x^n - 1$, em que *n* é um número primo. Então, para um dado problema de dimensão de reticulados d, é possível instanciar o sistema com $n = d$ ou $n = 1$. Se $n = d$, então, temos uma LWR EM ANEL DE NTRU e se $n = 1$, então, há um esquema de LWR.

[100] Note-se que até mesmo se a descrição usar como parâmetros de entrada (d,n) ou (d,d) ou (d,1), pode-se ter também uma configuração ainda mais geral na qual os parâmetros de entrada são (k*n,n) em que n pode ser ou 1 ou um número primo (no caso do Anel de NTRU).

Outras modalidades:

[101] Constatou-se que a descrição detalhada não depende do tipo de segredo ou do fato de usar arredondamento ou adição padrão de ruído (seja uma distribuição Gaussiana, uniforme ou qualquer outra). Desse modo, a presente invenção pode ser implementada para qualquer tipo de distribuição secreta.

[102] **Aplicabilidade à RLWE:** As informações adicionais a seguir contêm específicas para o anel de NTRU e considera p e q como potência de dois. Essas escolhas são

motivadas pelo tipo específico do método de reconciliação e pelo fato de que tendo p e q como potência de dois causa a uma implementação mais eficiente. As modalidades alternativas aplicam as ideias no presente documento a R/LWE. Nesse caso, é necessário adicionar ruído em vez de aplicar arredondamento.

[103] **Aplicabilidade a outros protocolos:** Neste documento foi mostrada a maneira de aplicar o presente modelo |à construção de KEX, KEM ou PKE. A mesma metodologia funciona também tipos de esquemas similares a El Gamal. O esquema também pode ser aplicável a outros esquemas, como assinaturas.

Utilização de outros anéis

[104] A maioria das modalidades neste documento são baseadas no anel NTRU $f(x) = x^n - 1$. Entretanto, esse polinômio não é irredutível e é igual a (x-1)(x^(n-1)+x^(n-2)+…+x+1. Isso facilita solucionar o problema de decisão de RLWE (b=as + e). No entanto, encontrar o s ainda é difícil.

[105] A literatura usa outros anéis que podem ser usados nas modalidades acima. Por exemplo, é possível usar anéis ciclotômicos $x^n + 1$ em que n é uma potência de dois e q é um número primo e $q \equiv 1(mod\, n)$. É possível também usar anéis ciclotômicos da forma $f(x) = x^{n-1} + x^{n-2} + \cdots + x + 1$ e q um número primo e $q \equiv 1(mod\, n)$.

[106] É possível usar esses anéis de maneira semelhante, conforme mostrado acima para o anel de NTRU. Por exemplo, o caso de não anel (LWE) é, então, implementado om o anel $(x^{n=1} + 1)$ ou (1) quando $n = 1$ para os anéis ciclotômicos. Note-se que em ambos os casos, q é um número primo. Note-se que o método de reconciliação é diferente do método usado na modalidade detalhada acima, visto que é necessário que q seja

uma potência de dois. Uma escolha adequada é uma reconciliação de chave como a reconciliação em Frodo.

[107] Outra boa opção é usar polinômios ciclotômicos como os supracitados, a saber, $x^n + 1$ e $x^{n-1} + x^{n-2} + \cdots + x + 1$ e usar um q que é uma potência de dois em vez de ser um número. A exigência em relação a q que é número primo principalmente é usar principalmente a NTT, no entanto, para uma implementação do presente esquema, a escolha de q para ser uma potência de dois é melhor visto que uma boa opção é implementar o esquema por meio de multiplicações de matriz.

[108] Outra boa opção é usar polinômios ciclotômicos como os supracitados, a saber, $x^n + 1$ e $x^{n-1} + x^{n-2} + \cdots + x + 1$ e usar um q que é uma potência de dois com a exigência adicional de que o polinômio seja um módulo irredutível 2.

[109] Note-se também caso o polinômio ciclotômico primário seja usado, $x^{n-1} + x^{n-2} + \cdots + x + 1$, então, as operações ainda podem ser executadas no anel x^n-1, uma vez que ambos os anéis diferem apenas por um termo x-1. Para esta finalidade, é necessário apenas para levantar os valores de um anel para outro multiplicando-se por x-1. Quando as operações são realizadas, é possível voltar para o anel anterior dividindo-se pela mesma quantidade.

Base lógica de escolhas de parâmetro de um ponto de vista de desempenho

[110] Algumas otimizações computacionais conhecidas para RLWE não podem ser aplicadas a algumas a algumas modalidades apresentadas neste documento. Em particular, não é possível aplicar a NTT ao anel de NTRU. Uma alternativa seria usar parâmetros adequados à NTT, como n uma potência de 2 e q números primos de modo que as operações em

Zq[x]/f(x), em que f(x)=x^n+1, possam ser realizadas por meio da NTT. Essa otimização acelerará desempenho da CPU para o caso de anel, (1>n) devido ao uso da NTT, no entanto, é esperado um pior desempenho um caso de não anel (n=1), uma vez que as operações seriam módulo q (em que q é um número primo).

[111] As escolhas de parâmetro apresentadas neste documento parecem ser as melhores para um esquema contido, visto ele permite uma implementação rápida de operações quando realizadas por meio de operações de matriz/vetor aplicáveis a qualquer escolha de n. Em outras palavras, até mesmo caso o esquema para n>1 não seja eficiente quanto seria outro esquema se anel que usa NTT, a escolha neste documento permite uma implementação muito rápida do esquema quando n>1, mas também quando n=1. Além disso, o uso de anéis de NTRU permite um ajuste fino de exigências de segurança e de largura de banda visto que há muitos anéis adequados.

Parâmetros

[112] O esquema pode ser configurado com diferentes parâmetros. Os valores de d e q e p definem a dificuldade do problema de reticulado subjacente. Valores exemplificativos são, por exemplo, um d de aproximadamente 700 e q e p iguais a 2^14 e 2^11. Nesse caso, n_bar e m_bar podem ser iguais a 8 de modo que 4 bits possam ser obtidos de cada coeficiente para a matriz de chave resultante em uma KEX quando n=1. Quando n=d, então, é necessário um único bit por coeficiente de polinômio de modo que o q possa ser menor e, portanto, também o p e, portanto, também o n. Uma vez que um

polinômio tem n coeficientes, então, uma chave de n bits é obtida e n_bar e m_bar precisam ser iguais apenas a um.

[113] Constata-se que para algumas escolhas de anéis, é possível ter múltiplos conjuntos de parâmetros. Por exemplo, considerando os polinômios ciclotômicos primos, pode-se ter as seguintes classes de parâmetros:

[114] - Parâmetros para n=1 e n=d que determinam se o problema subjacente é baseado ou não em uma estrutura de anel.

[115] - q é um número primo ou potência de dois determinar o tipo de otimizações que são viáveis de um ponto de vista de CPU.

Configurabilidade do esquema

[116] O esquema pode ser configurado para diferentes níveis de segurança e para um problema difícil subjacente de acordo com o seguinte:

[117] Primeiramente, para cada um dos problemas subjacentes (por exemplo, LWR ou RLWR), um programa de computador computa os parâmetros-alvo (principalmente, d, n, q e p) para alcançar um determinado nível de segurança e desempenho em termos de probabilidade de falha, largura de banda e CPU. Essa tarefa pode ser executada uma única vez e, em seguida, os parâmetros podem ser publicados.

[118] Em segundo lugar, os conjuntos de parâmetros são publicados são armazenados em uma tabela junto do programa que implementar o esquema proposto neste documento, sendo que cada conjunto de parâmetros correspondem a um problema subjacente, nível de segurança e parâmetros de desempenho.

[119] Em terceiro lugar, quando duas partes (por exemplo, Alice e Bob) desejam se comunicar, Alice informa a Bob sobre o conjunto de parâmetros que ela está disposta a usar. Isso significa que Alice pode determinar se ela quer usar o esquema, por exemplo, em modo de anel ou em modo de não anel (escolhas d e n) e nesse nível de segurança (dependente de d, q e p). Alice pode informar a Bob sobre sua escolha especificando-se explicitamente os parâmetros definidos (d, n, q, p, …), enviando-se um identificador do conjunto de parâmetros ou especificando-se valores de desempenho (por exemplo, tamanho da chave pública, tamanho da chave privada...) que dependem dos parâmetros escolhidos. Note-se que Bob pode também então ter uma política que exige não apenas um nível mínimo de segurança, mas sim também um certo problema difícil. Desse modo, caso a proposta de Alice não seja satisfatória, então, Bob pode solicitar um conjunto diferente de parâmetros.

[120] Abaixo, encontra-se mais informações sobre o supracitado, e modalidades adicionais, de um ponto de vista matemático.

1 Especificações de algoritmo e documentação de apoio

1.1 Base lógica de projeto

1.2 Problemas subjacentes

1.3 Preliminares

1.4 Criptografia de chave pública de CPA

1.5 Método de encapsulação de chave de CPA para o caso de não anel

1.6 Estrutura de método de encapsulação de chave de CCA para o caso de não anel

1.7 Caso de anel e de não anel combinados

1.7.1 Configurações de estrutura

1.7.2 Conjuntos de parâmetro

1.7.3 Níveis de NIST

1 Especificações de algoritmo e documentação de apoio

1.1 Base lógica de projeto

[121] Algumas modalidades se referem a uma estrutura de algoritmos para possibilitar a encapsulação de chave. Os algoritmos propostos se enquadram na categoria de criptografia baseadas em reticulados, em particular, eles podem ser baseados no problema de Aprendizagem com Arredondamento (LWR). Esse problema foi escolhido para o projeto dos algoritmos devido ao seu extenso estudo e suas propriedades vantajosas de desempenho, por exemplo, exigências de largura de banda reduzida.

[122] Uma característica interessante de algumas modalidades é que o fato de que a estrutura foi projetada para instanciar o problema de LWR e o LWR em Anel (RLWR) de maneira contínua. Isso significa que os mesmos algoritmos (e código) podem ser usados a fim de instanciar eficientemente LWR- ou algoritmos baseados em RLWR. A base lógica dessa escolha é tem vários pontos.

• Primeiramente, isso permite a adaptação a múltiplos ambientes de maneira simples: por um lado, os algoritmos baseados em LWR podem ser aplicados a ambientes em que o desempenho é menos de um problema, porém a segurança é a prioridade de modo que seja preferencial não ter a estrutura de anel adicional. Por outro lado, os algoritmos de RLWR baseados em alcançar o melhor desempenho em termos de largura de banda e computação de modo que possam melhor se adaptar a ambientes mais limitados.

• Em segundo lugar, essa abordagem possibilita que uma estratégia de migração de esquemas baseados em anel para

esquemas de não anel no primeiro dia. Isso garante que caso ataques aos esquemas de anel sejam encontrados futuramente, devido à estrutura de anel subjacente, então, uma solução segura alternativa já está disponível e instalada.

• Em terceiro lugar, o quadro proposto reduz a manutenção e a análise de códigos, visto que a mesma implementação instancia os casos de RLWR e LWR para todos os algoritmos CPA KEM, CCA KEM e CCA PKE.

[123] A modalidade otimizar adicionalmente o desempenho por meio de diversos recursos de diferenciação. Em primeiro lugar, o uso de LWR e RLWR otimiza o desempenho de largura de banda e evita rotinas de amostragem. Em segundo lugar, segredos ternários segredos escassos são escolhidos visto que eles facilitam a implementação, permitem desempenho aprimorado e probabilidades de falha. Em terceiro lugar, a fim de impedir ataques de pré-computação, a matriz pública A não é mantida constante, porém é renovada de maneira muito eficiente para minimizar degradação de desempenho devido à sua recomputação durante o handshake. Em quarto lugar, a funcionalidade para realizar o arredondamento, compressão de mensagem e descriptografia é instanciada por meio da mesma função. Isso simplifica e otimiza o projeto. Em quinto lugar, a versão de anel dessa proposta se baseia no anel de NTRU devido a seus recursos de bom desempenho.

## 1.3 Preliminares

[124] O anel de polinômio $\mathbb{Z}[x]/(x^n-1)$ é denotado por $R_n$. Será permitido que $n=1$; então, $R_n=\mathbb{Z}$. Como abuso de notação, para cada número inteiro positivo $a$, é escrito $\mathbb{Z}_a$ para o conjunto $\{0,1,\ldots,a-1\}$, e $R_{n,a}$ para o conjunto de polinômios

de grau menor que $n$ com todos os coeficientes em $\mathsf{Z}_a$. Um polinômio em $R_n$ é chamado de *trinário* caso todos os seus coeficientes sejam 0, 1 ou $-1$.

[125] Ao longo deste documento, letras de fonte regular denotam elementos de $R_n$, e letras minúsculas em negrito representam vetores com coeficientes em $R_n$. Todos os vetores são vetores de coluna. Letras maiúsculas em negrito são matrizes. A transposição de um vetor $\mathbf{v}$ ou uma matriz $\mathbf{A}$ é denotada por $\mathbf{v}^T$ ou $\mathbf{A}^T$.

[126] **Reduções modulares** Para um número inteiro positivo $\alpha$ e $x \in \mathsf{Q}$, define-se $\{x\}_\alpha$ como o elemento exclusivo $x'$ no intervalo $(-\alpha/2, \alpha/2]$ que satisfaz $x' \equiv x \pmod{\alpha}$. Além disso, define-se $\langle x \rangle_\alpha$ como o único elemento $x'$ no intervalo $[0, \alpha - 1]$ de modo que $x \equiv x' \pmod{\alpha}$.

[127] **Arredondamento** Para $x \in \mathsf{Q}$, denota-se por $\lceil x \rfloor$ o arredondamento para o número inteiro mais próximo de $x$, com arredondamento para cima no caso de empate.

[128] **Compressão e descompressão** Caso $a, b$ seja números inteiros de modo que $a > b$. As funções Comprimir $_{a,b}: \mathsf{Z}_a \to \mathsf{Z}_b$ e Descomprimir $_{a,b}: \mathsf{Z}_b \to \mathsf{Z}_a$ são definidas como

$$\mathit{Comprimir}_{a,b}(x) = \langle \lceil \frac{b}{a} \cdot x \rfloor \rangle_b, \text{ e } \mathit{Descomprimir}_{a,b}(x) = \langle \lceil \frac{a}{b} \cdot x \rfloor \rangle_a$$

[129] Pode ser mostrado que Descomprimir $_{a,b}$ é "quase" o inverso de Comprimir $_{a,b}$. Para ser mais preciso, para cada $x \in \mathsf{Z}_a$,

$$|\{\mathit{Descomprimir}_{a,b}(\mathit{Comprimir}_{a,b}(x)) - x\}_a| \leq \frac{a+1}{2b}.$$

[130] Por meio de computação direta, pode-se observar que caso $a$ seja um múltiplo de $b$, então

$$Descomprimir_{a,b}(Comprimir_{a,b}(x)) \equiv x + \{x\}_{b/a} (\mathrm{mod}\, a)$$

para cada $x \in \mathsf{Z}_a$.

A função comprimir serve para três propósitos.

• Primeiramente, aprimora a segurança visto que leva ao problema de Aprendizado com Arredondamento.

• Em segundo lugar, ela é usada para descriptografar uma mensagem.

• Em terceiro lugar, ela reduz os tamanhos de chaves públicas e textos cifrados reduzindo, então, as exigências de banda de largura.

[131] Para cada um $v \in R_n$, o **peso de Hamming** de $v$ é definido como seu número de coeficientes diferentes de zero. O peso de Hamming de um vetor em $R_n^k$ é igual à soma do peso de Hamming de seus componentes. Com $H_{n,k}(h)$ é denotado o conjunto de todos os vetores $\mathbf{v} \in R_n^k$ polinômios trinários do peso de Hamming $h$. Considerando-se os coeficientes de um polinômio em $R_n$ como um vetor de comprimento $n$, um polinômio em $H_{n,k}(h)$ correspondente a um vetor trinário de comprimento $nk$ com não zeros nas $h$ posições, de modo que $H_{n,k}(h)$ tenha elementos $\binom{nk}{h} 2^h$.

[132] Para $v = \sum_{i=0}^{n-1} v_i x^i \in R_n$, define-se $\| v \|_\infty$ como o maior valor absoluto de seus coeficientes, ou seja,

$$\| v \|_\infty = \max\{ | v_i \| 0 \le i \le n-1 \}.$$

[133] É fácil ver que para todos $a, b \in R_n$, há

$$\|ab\|_\infty \le n \cdot \|a\|_\infty \|b\|_\infty .$$

[134] De modo similar, caso $\mathbf{V}$ seja uma matriz de polinômios em $R_n$, então, $\|\mathbf{V}\|_\infty$ é o maior que todos os coeficientes dos polinômios em $\mathbf{V}$. Por fim, para $\mu \le ab$ e $\mathbf{C} \in Z^{a \times b}$, o vetor Amostra $_{\mu,a,b}(\mathbf{C}) \in Z_p^\mu$ compreende os valores nas porções pré-especificadas $\mu$ do $\mathbf{C}$.

1.4 Criptografia de chave pública de CPA

[135] Nessa seção, descrevemos Criptografia de chave pública para o caso sem anel. Nesse esquema de criptografia de chave pública seguro de CPA é descrito com os algoritmos abaixo. Esses algoritmos assumem o conhecimento de vários parâmetros de sistema, *como* números inteiros positivos $n, h, p, q, t, B, \bar{n}, \bar{m}, \mu$. Os algoritmos incluem escolhas aleatórias, para uma matriz $\mathbf{A} \in Z_q^{n \times n}$, assim como para uma $n \times \bar{n}$ matriz $\mathbf{S}$ e para uma $n \times \bar{m}$ matriz $\mathbf{R}$, sendo que ambas têm todas as colunas $T_n(h)$. Ou seja, tanto $\mathbf{S}$ quanto $\mathbf{R}$ são matrizes trinárias para as quais cada coluna tem peso $h$ de Hamming. Para fazer a escolha para $\mathbf{A}$ explícito, os parâmetros de sistema incluem um espaço de aleatoriedade $\Sigma$ e um mapeamento $f : \Sigma \to Z_q^{n \times n}$. De modo similar, para tornar explícitas as escolhas para $\mathbf{S}$ e $\mathbf{R}$, são definidas as funções $f_S$ e $f_R$ que atribuem, em uma entrada de $\{0,1\}^{r_S}$ e $\{0,1\}^{r_R}$, respectivamente, gerando $n \times \bar{n}$ e $n \times \bar{m}$ trinários, respectivamente, a cada colune que tem peso de Hamming $h$. Para um conjunto $A$, é denotado por $a \xleftarrow{\$} A$ que $a$ é extraído uniformemente de $A$.

[136] O primeiro algoritmo gera uma chave pública $pk$ em $\Sigma \times Z_p^{\bar{n}}$, e uma chave secreta $sk \in T_n(h)^{\bar{n}}$, isto é, uma matriz $n \times \bar{n}$ trinária da qual cada coluna tem peso de Hamming $h$.

[137] Algoritmo 1: PST.NR.CPA-PKE.Keygeneration()

[138] Números inteiros de parâmetros $p, q, n, h, \bar{n}$

[139] Entrada -

[140] Saída $pk \in \Sigma \times Z_p^{n \times \bar{n}}, sk \in (T_n(h))^{\bar{n}}$

$\sigma \xleftarrow{\$} \Sigma$

$\mathbf{A} = f(\sigma)$

$s \xleftarrow{\$} \{0,1\}^{r_S}$

$\mathbf{S} = f_S(s)$

$\mathbf{B} = \text{Comprimir}_{q,p}(\langle \textbf{Sinal de Ativaēćo (AS)} \rangle_q)$.

$pk = (\sigma, \mathbf{B})$

$sk = \mathbf{S}$

**retornar** $(pk, sk)$

[141] O próximo algoritmo gera, a partir de uma chave pública $pk$, uma mensagem e uma variável de arredondamento $m$ $\rho$ um texto cifrado $c$.

[142] Algoritmo 2: PST.NR.CPA-PKE.Encryptwithrho( $pk, m, \rho$ )

[143] Números inteiros de parâmetros $p, t, q, n, \bar{m}, \bar{n}, \mu$

[144] Entrada $pk = (\sigma, \mathbf{B}) \in \Sigma \times Z_p^{n \times \bar{n}}, m \in Z_{2^B}^{\mu}, \rho \in \{0,1\}^{r_R}$

[145] Saída $c = (\mathbf{U}, \mathbf{v}) \in Z_p^{n \times \bar{m}} \times Z_t^{\mu}$

$\mathbf{A} = f(\sigma)$

$\mathbf{R} = f_R(\rho)$

$\mathbf{U} = Comprimir_{q,p}(\langle \mathbf{A}^T \mathbf{R} \rangle_q)$

$\mathbf{v} = Comprimir_{p,t}(\langle Amostra_{\mu}(\mathbf{B}^T \mathbf{R}) + \lceil \frac{p}{2^B} \rceil \cdot m \rangle_p)$

$c = (\mathbf{U}, \mathbf{v})$

**retornar** $c$

[146] O algoritmo de criptografia gera uma chave pública $pk$ e de uma mensagem $m$ um texto cifrado $c$.

[147] Algoritmo 3: PST.NR.CPA-PKE.Encrypt( $pk, m$ )

[148] Números inteiros de parâmetros $p, t, q, n, \bar{m}, \bar{n}, \mu$

[149] Entrada $pk = (\sigma, \mathbf{B}) \in \Sigma \times Z_p^{n \times \bar{n}}, m \in Z_{2^B}^{\mu}$

[150] Saída $c = (\mathbf{U}, \mathbf{v}) \in Z_p^{n \times \bar{m}} \times Z_t^{\mu}$

$\rho \xleftarrow{\$} \{0,1\}^{r_R}$

$c =$ PST.NR.CPA-PKE.Encryptwithro( $pk, m, \rho$ )

**retornar** $c$

[151] O algoritmo de descriptografia recupera, de uma chave secreta $sk$ e um texto cifrado $c$, uma (estimativa para) a mensagem de texto $m$.

[152] Algoritmo 4: PST.NR.CPA-PKE.Decrypt( $sk, c$ )

[153] Números inteiros de parâmetros $p, t, q, n, \bar{m}, \bar{n}, \mu$

[154] Entrada $sk = \mathbf{S} \in (T_n(h))^{\bar{n}}, c = (\mathbf{U}, \mathbf{v}) \in Z_p^{n \times \bar{m}} \times Z_t^{\mu}$

[155] Saída $\hat{m} \in Z_{2^B}^{\mu}$

$\mathbf{v} = Descomprimir_{p,t}(\mathbf{v})$

$\hat{m} = Comprimir_{p,2^B}(\mathbf{v} - Amostra_{\mu}(\mathbf{S}^T \mathbf{U}))$

**retornar** $\hat{m}$

**Nota** A operação do esquema poderia ser explicada aqui.

1.5 Método de encapsulação de chave de CPA para o caso de não anel

[156] Os três algoritmos a seguir são construídos com base no esquema de criptografia de chave pública definido em uma seção anterior para definir um método de encapsulação de chave de CPA segura. Isso emprega uma função de hash $H:\{0,1\}^{*}\rightarrow\{0,1\}^{\mu B}$.

[157] **Nota:** Pode-se ter, também, uma função de hash adicional $G:\{0,1\}^{*}\rightarrow\{0,1\}^{\kappa}$, e aplicar $G$ em vez de $H$ nas etapas 4 do algoritmo de encapsulação e na etapa 2 do algoritmo de desencapsulação.

[158] Algoritmo 5: PST.CPA-KEM.Keygeneration()

[159] Números inteiros de parâmetros $p,q,n,h,\bar{n}$

[160] Entrada

[161] Saída $pk\in\Sigma\times Z_p^{n\times\bar{n}}, sk\in(T_n(h))^{\bar{n}}$

$(pk,sk)$ = PST.NR.CPA-PKE.Keygeneration()

**retornar** $(pk,sk)$

[162] Algoritmo 6: PST.CPA-KEM.Encapsulate( $pk$ )

[163] Números inteiros de parâmetros $p,t,q,n,\bar{m},\bar{n},\mu,\alpha$

[164] Entrada $pk=(\sigma,\mathbf{B})\in\Sigma\times Z_p^{n\times\bar{n}}$

[165] Saída $c=(\mathbf{U},\mathbf{v})\in Z_p^{n\times\bar{m}}\times Z_t^{\mu}$, $K\in Z_{2^B}^{\mu}$

$a\xleftarrow{\$}\{0,1\}^{\alpha}$

$m=H(a,pk)$

$c$ = PST.NR.CPA-PKE.Encrypt( $pk,m$ )

$K=H(m,c)$

**retornar** $(c,K)$

[166] Algoritmo 7: PST.NR.CPA-KEM.Decapsulate($sk, c$)

[167] Números inteiros de parâmetros $p, t, q, n, \bar{m}, \bar{n}, \mu$

[168] Entrada $sk = \mathbf{S} \in (T_n(h))^{\bar{n}}, c = (\mathbf{U}, \mathbf{v}) \in \mathbb{Z}_p^{n \times \bar{m}} \times \mathbb{Z}_t^{\mu}$

[169] Saída $K \in \mathbb{Z}_{2^B}^{\mu}$

$m$ = PST.NR.CPA-PKE.Decrypt($sk, c$)

$K = H(m, c)$

**retornar** $K$

[170] **Observação** Deve-se notar que nos Algoritmos 1.5 e 1.5, $pk$ e $c$ são mapeados implicitamente para uma cadeia de bits antes de alimentar $H$ com os mesmos.

1.6 Estrutura de método de encapsulação de chave de CCA para o caso de não anel

[171] A estrutura de encapsulação de chave é obtida mediante a aplicação de uma variante de KEM da transformada de Fujisaki-Okamoto para o presente esquema de criptografia para torná-lo seguro em termos de CCA.

[172] Foi usada a notação do esquema de CPA-PKE. É necessário também duas funções de hash, $G: \{0,1\}^* \rightarrow \{0,1\}^{\beta} \times R_R \times \{0,1\}^{\gamma}$ e $H: \{0,1\}^* \rightarrow K = \{0,1\}^{\kappa}$.

[173] Primeiramente, é descrito o algoritmo de geração de chave. Note-se que, no caso de escolhas iguais de aleatorização, as chaves públicas $pk$ são iguais; a chave secreta com o Algoritmo 1.7 compreende a chave secreta, bem como a chave pública e uma cadeia aleatória ($z$).

[174] Algoritmo 8: PST.NR.CCA-KEM.Keygen()

[175] Números inteiros de parâmetros $p, q, n, h, \bar{n}, \zeta$

[176] Entrada

[177] Saída – $pk \in \Sigma \times Z_p^{n \times \bar{n}}, sk \in (T_n(h))^{\bar{n}} \times \{0,1\}^{\zeta}$

$(pk_{CPA-PKE}, sk_{CPA-PKE}) =$ PST.NR.CPA-PKE.Keygen()

$z \xleftarrow{\$} \{0,1\}^{\zeta}$

$pk = pk_{CPA-PKE}$

$sk = (sk_{CPA-PKE}, z)$

**retornar** $(pk, sk)$

[178] A saída $c$ do Algoritmo 1.6 compreende, com entrada igual à do Algoritmo 1.4 e a mesma escolha para $\mathbf{R}$, o texto cifrado $(\mathbf{U}, \mathbf{v})$ do Algoritmo 1,4. Ela também compreende uma chave $K$.

[179] Algoritmo 9: PST.NR.CCA-KEM.Encapsulate( $pk$ )

[180] Números inteiros de parâmetros $p, t, q, n, \bar{m}, \bar{n}, \mu$

[181] Entrada $pk = (\sigma, \mathbf{B}) \in \Sigma \times Z_p^{n \times \bar{n}}$

[182] Saída $c = (\mathbf{U}, \mathbf{v}, d) \in Z_p^{n \times \bar{m}} \times Z_t^{\mu} \times \{0,1\}^{\gamma}$

$m \xleftarrow{\$} Z_{2^B}^{\mu}$

$(\hat{K}, \rho, d) = G(\sigma, \mathbf{B}, m)$

$(\mathbf{U}, \mathbf{v})$ = PST.CPA.PKE.Encryptwithro( $pk, m, \rho$ )

$c = (\mathbf{U}, \mathbf{v}, d)$

$K = H(\hat{K}, c)$

**retornar** $(c, K)$

[183] Com chave secreta igual $\mathbf{S}$ e entrada igual $(\mathbf{U}, \mathbf{v})$, o valor de $m'$ é igual àquele fornecido ao Algoritmo 1.4. Conclui-se que caso as escolhas aleatórias nos algoritmos NR CCA sejam iguais àquelas dos algoritmos NRCPA PKE e caso os algoritmos de NRCPA-PKE recuperem corretamente a mensagem $m$

escolhida na Linha 1 do Algoritmo 1.6, então $m = m'$. Nesse caso, $(\hat{K}', \rho', d')$, conforme computado da Linha 2 do Algoritmo 1.6, igual a $(\hat{K}, \rho, d)$ na Linha 2 do Algoritmo 1.6 e, portanto, os valores de $K$, conforme computado na Linha 1.6 do Algoritmo 10 e do Algoritmo 1.6 são iguais. Caso a condição na Linha 7 do Algoritmo 1.6 não seja satisfeita, então, a saída é uma chave aleatória.

[184] Algoritmo 10: PST.NR.CCA-KEM.Decapsulate( $pk, sk, c$ )

[185] Números inteiros de parâmetros $p, t, q, n, \overline{m}, \overline{n}, \mu, \zeta$

[186] Entrada

$pk = (\sigma, \mathbf{B}) \in \Sigma \times \mathsf{Z}_p^{n \times \overline{n}}, sk = (\mathbf{S}, z) \in (T_n(h))^{\overline{n}} \times \{0,1\}^{\zeta}, \; c = (\mathbf{U}, \mathbf{v}, d) \in \mathsf{Z}_p^{n \times \overline{m}} \times \mathsf{Z}_t^{\mu} \times \{0,1\}^{\gamma}$

[187] Saída $K \in K$

$m' = PST.NR.CPA - PKE.Decrypt(\mathbf{S}, (\mathbf{U}, \mathbf{v}))$

$(\hat{K}', \rho', d') = G(\sigma, \mathbf{B}, m')$

$(\mathbf{U}', \mathbf{v}')$ = PST.CPA-PKE.Encryptwithrho$(pk, m', \rho')$

**se** $(\mathbf{U}', \mathbf{v}', d') = (\mathbf{U}, \mathbf{v}, d)$ **então**

**retornar** $K = H(\hat{K}', c)$

**senão**

**retornar** $K = H(z, c)$

**end if**

[188] **Observação** Nos Algoritmos 1.6 e 1.6, mapeamos implicitamente $(\sigma, \mathbf{B})$ $m'$ e $c$ para cadeias binárias antes de alimentar $G$ e $H$ com as mesmas, respectivamente.

1.7 Caso de anel e de não anel combinados

[189] Nessa seção, é descrita versão de não anel e de anel combinadas do algoritmo anterior, com anel polinômio $R_n = \mathbb{Z}[x]/(x^n - 1)$. É usado $*$ para denotar a multiplicação em $R_n$. A fim de salientar a similaridade com o caso de não anel, o polinômio $\sum_{i=0}^{n-1} f_i x^i$ e o vetor (coluna) $(f_0, f_1, \ldots, f_{n-1})^T$ são identificados. Para $\mathbf{f}, \mathbf{g} \in \mathbb{Z}^n$, $\mathbf{f} * \mathbf{g}$ é definido como o vetor de coeficiente do polinômio $f(x)g(x)(\mathrm{mod}(x^n - 1))$. Ou seja, para $0 \le i \le n-1$,

$$(\mathbf{f} * \mathbf{g})_i = \sum_{j=0}^{n-1} f_j g_{\langle i-j \rangle_n}.$$

[190] É definida também uma multiplicação $*$ de matrizes, conforme a seguir. Caso $\mathbf{B} \in \mathbb{Z}^{n \times \bar{n}}$ e caso $\mathbf{R} \in \mathbb{Z}^{n \times \bar{m}}$, por exemplo

$$\mathbf{B} = [\mathbf{b}_0, \ldots, \mathbf{b}_{\bar{n}-1}] \quad e \quad \mathbf{R} = [\mathbf{r}_0, \ldots, \mathbf{r}_{\bar{m}-1}].$$

[191] A matriz $n\bar{n} \times \bar{m}$ de $\mathbf{B}^T * \mathbf{R}$ é definida como

$$\mathbf{B}^T * \mathbf{R} = \begin{bmatrix} \mathbf{b}_0 * \mathbf{r}_0 & \mathbf{b}_0 * \mathbf{r}_1 & \cdots & \mathbf{b}_0 * \mathbf{r}_{\bar{m}-1} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{b}_{\bar{n}-1} * \mathbf{r}_0 & \mathbf{b}_{\bar{n}-1} * \mathbf{r}_1 & \cdots & \mathbf{b}_{\bar{n}-1} * \mathbf{r}_{\bar{m}-1} \end{bmatrix}$$

[192] Para distinguir entre o caso de anel e o caso de não anel, é usado um parâmetro *anel* Booleano.

[193] Algoritmo 11: PST.CPA-PKE.Keygeneration()

[194] Números inteiros de parâmetros $p, q, n, h, \bar{n}$; *anel* Booleano **Requisito** If *anel* then $\mu \le n\bar{n} \cdot \bar{m}$ else $\mu \le \bar{n} \cdot \bar{m}$

[195] Entrada

[196] Saída - $pk \in \Sigma \times \mathbb{Z}_p^{n \times \bar{n}}, sk \in (T_n(h))^{\bar{n}}$

$s \xleftarrow{\$} \{0,1\}^{r_s}$

$\mathbf{S} = f_S(s)$

$\sigma \xleftarrow{\$} \Sigma$

**if** *anel*

**então, começar** $\mathbf{a} = f_{pol}(\sigma); \mathbf{B} = \langle \mathbf{a} * \mathbf{S} \rangle_q$ **fim**

**senão, começar** $\mathbf{A} = f(\sigma); \mathbf{B} = (\langle$ **Sinal de Ativaēćo (AS)** $\rangle_q)$ **fim**

$\mathbf{B} = \text{Comprimir}_{p,q}(\mathbf{B})$

$pk = (\sigma, \mathbf{B})$

$sk = \mathbf{S}$

**retornar** $(pk, sk)$

[197] O próximo algoritmo gera, a partir de uma chave pública $pk$, uma mensagem e uma variável de arredondamento $m$ $\rho$ um texto cifrado $c$.

[198] Algoritmo 12: PST.CPA-PKE.Encryptwithrho( $pk, m, \rho$ )

[199] Números inteiros de parâmetros $p, t, q, n, \bar{m}, \bar{n}, \mu$ ; *anel* Booleano **Requisito** If *anel* then $\mu \leq n\bar{n} \cdot \bar{m}$ else $\mu \leq \bar{n} \cdot \bar{m}$

[200] Entrada $pk = (\sigma, \mathbf{B}) \in \Sigma \times Z_p^{n \times \bar{n}}, m \in Z_{2^B}^{\mu}, \rho \in \{0,1\}^{r_R}$

[201] Saída $c = (\mathbf{U}, \mathbf{v}) \in Z_p^{n \times \bar{m}} \times Z_t^{\mu}$ $\mathbf{R} = f_R(\rho)$

**if** *anel*

**então, começar** $\mathbf{a} = f_{pol}(\sigma); \mathbf{U} = \langle \mathbf{a} * \mathbf{R} \rangle_q$ **fim**

**senão, começar** $\mathbf{A} = f(\sigma); \mathbf{U} = \langle \mathbf{A}^T \mathbf{R} \rangle_q$ **fim**

$\mathbf{U} = Comprimir_{q,p}(\mathbf{U})$

**if** *anel*

**então** $\mathbf{V} = Amostra_{\mu, n\bar{n}, \bar{m}}(\langle \mathbf{B}^T * \mathbf{R} \rangle_p)$

**senão** $\mathbf{V} = Amostra_{\mu, \bar{n}, \bar{m}}(\langle \mathbf{B}^T \mathbf{R} \rangle_p)$ ;

$\mathbf{v} = Comprimir_{p,t}(\langle \mathbf{v} + \lceil \frac{p}{2^B} \rceil \cdot m \rangle_p)$

$c = (\mathbf{U}, \mathbf{v})$

**retornar** $c$

[202] O algoritmo de criptografia gera uma chave pública $pk$ e de uma mensagem $m$ um texto cifrado $c$.

[203] Algoritmo 13: PST.CPA-PKE.Encrypt( $pk, m$ )

[204] Números inteiros de parâmetros $p, t, q, n, \bar{m}, \bar{n}, \mu$, *anel* Booleano

[205] Entrada $pk = (\sigma, \mathbf{B}) \in \Sigma \times Z_p^{n \times \bar{n}}, m \in Z_{2^B}^{\mu}$

[206] Saída $c = (\mathbf{U}, \mathbf{v}) \in Z_p^{n \times \bar{m}} \times Z_t^{\mu}$

$\rho \xleftarrow{\$} \{0,1\}^{r_R}$

$c =$ PST.CPA-PKE.Encryptwithro( $pk, m, \rho$)

**retornar** $c$

[207] O algoritmo de descriptografia recupera, de uma chave secreta $sk$ e um texto cifrado $c$, uma (estimativa para) a mensagem de texto $m$.

[208] Algoritmo 14: PST.CPA-PKE.Decrypt( $sk, c$ )

[209] Números inteiros de parâmetros $p, t, q, n, \bar{m}, \bar{n}, \mu$; Booleano: *anel* **Requisito** If *anel* then $\mu \leq n\bar{n} \cdot \bar{m}$ else $\mu \leq \bar{n} \cdot \bar{m}$

[210] Entrada $sk = \mathbf{S} \in (T_n(h))^{\bar{n}}, c = (\mathbf{U}, \mathbf{v}) \in Z_p^{n \times \bar{m}} \times Z_t^{\mu}$

[211] Saída $\hat{m} \in Z_{2^B}^{\mu}$

$\mathbf{v} =$ Descomprimir $_{p,t}(\mathbf{v})$

**if** *anel*

**então** $X = Amostra_{\mu, n\bar{n}, \bar{m}}(\langle \mathbf{S}^T * \mathbf{U} \rangle_p)$

**senão** $X = Amostra_{\mu,n,m}(\langle S^T \mathbf{U} \rangle_p)$

$\hat{m} = Comprimir_{p,2^B}(\mathbf{v} - \mathbf{X})$

**retornar** $\hat{m}$

1.7.1 Configurações de estrutura

[212] Os algoritmos acima permitem habilitar vários algoritmos que dependem de parâmetros de configuração.

- **CPA-KEM baseada em LWR** Para n=1 e k=d.
- **CPA-KEM baseada em RLWR** Para n=d e k=1.
- **CCA-KEM baseada em LWR** Para n=1 e k=d.
- **CCA-KEM baseada em RLWR** Para n=d e k=1.
- **CCA-KEM-DEM baseada em LWR** Para n=1 e k=d.
- **CCA-KEM-DEM baseada em RLWR** Para n=d e k=1.

Atualização da matriz compartilhada

[213] Acima, são fornecidas várias opções para atualizar uma matriz baseada em um grupo compartilhado. O grupo compartilhado pode assumir a forma de uma matriz mestre A_master ou $a_master$. O grupo compartilhado pode de fato ser representado como uma matriz A, mas isso não é necessário. A indexação do grupo compartilhado como matriz A torna determinadas permutações mais fáceis, por exemplo, permutações cíclicas de linhas ou colunas. De modo mais geral, o grupo compartilhado pode ser considerado com um grupo de dados nos quais a função de seleções pode atribuir elementos aos elementos na matriz compartilhada.

[214] A seguir, são fornecidas várias modalidades de grupos compartilhados:

[215] A matriz pública compartilhada ***A*** de dimensão $k \times k$ usada, por exemplo, em uma troca de chaves de LWE (e também, outras construções de LWE) pode ser computada

aplicando-se uma permutação $\boldsymbol{\pi}$ a uma matriz mestre $\boldsymbol{A_{master}}$ que é definida como tendo pelo menos a dimensão $k \times k$.

[216] Presumindo que $A_{master}$ é mantida na memória, essa abordagem permite uma maneira mais rápida de computar uma A nova. Isso é particularmente importante para servidores que têm de lidar com muitas conexões. Nas modalidades, $\boldsymbol{A}$ é computada a partir de $\boldsymbol{A_{master}}$ com tamanho menor que $k \times k$ aplicando-se múltiplas permutações. Desse modo, o tamanho de $\boldsymbol{A_{master}}$ é menor e múltiplas permutações são aplicadas no mesmo grupo. O último tem a vantagem de que um nó não precisa (i) computar toda a $A_{master}$ a partir de uma semente ou (ii) manter $A_{master}$ na memória, que é tão grande quanto a matriz compartilhada. Em vez disso, um nó poderia apagar uma matriz compartilhada após ter sido usado, na próxima vez que uma matriz compartilhada é necessária, ela pode ser construída a partir do grupo compartilhado. Em uma modalidade, o grupo compartilhado compreende apenas uma raiz quadrada na quantidade de dados que a própria matriz compartilhada precisa. Por exemplo, tal tamanho suporta o particionamento da matriz compartilhada em até $k$ de cada uma das $k$ entradas. Em uma modalidade, um grupo compartilhado compreende cerca de a mesma quantidade de entradas que o tamanho do primeiro módulo $q$, por exemplo, menor ou igual a $q$ ou $2q$ etc.

[217] Algumas modalidades almejam aprimorar o desempenho especialmente no lado do cliente (iniciador) em um KEX, visto que o iniciador tem de computar e/ou armazenar menos. Isso também aprimora o lado do servidor (respondente), mas esse aprimoramento é relativamente menor. Quando aplicado a outros esquemas baseados em reticulado, essa ID pode aprimorar o desempenho de esquemas de criptografia de chave pública ou

esquemas de assinatura, visto que o parâmetro público A também pode ser computado mais rapidamente quando recebe algumas sementes.

[218] A matriz pública compartilhada $\boldsymbol{A}$ de dimensão $k \times k$ usada em uma troca de chaves de LWE (e também, outras construções de LWE) pode ser computada aplicando-se uma permutação $\boldsymbol{\pi}$ a uma matriz mestre $\boldsymbol{A_{master}}$. Em uma modalidade, a matriz mestre pode ter uma dimensão de pelo menos $k \times k$. No entanto, a matriz compartilhada $\boldsymbol{A}$ também pode ser computado a partir de $\boldsymbol{A_{master}}$ com tamanho menor que $k \times k$ aplicando-se múltiplas funções, que podem ser permutações. O tamanho da $\boldsymbol{A_{master}}$ é, então, menor e múltiplas funções (permutações) são aplicadas no mesmo grupo. Presumindo que $A_{master}$ é mantida na memória, essa abordagem permite uma maneira mais rápida de computar uma A nova. Isso é particularmente importante para servidores que têm de lidar com muitas conexões.

Descrição geral:

[219] Uma matriz mestre $\boldsymbol{A_{master}}$ que contém $N$ elementos está disponível na memória, $N < k^2$. Os elementos na $\boldsymbol{A_{master}}$ podem ser computados a partir de uma semente aleatória $s$ aplicando-se uma função pseudoaleatória (PRF) a ela. A matriz mestre $\boldsymbol{A_{master}}$ é um exemplo de um grupo compartilhado.

[220] Em uma modalidade, a matriz $\boldsymbol{A}$ é particionada em grupos $B_i$, sendo que cada grupo tem o tamanho $N_B \leq N$. U grupo pode representar, por exemplo:

- Um bloco em $\boldsymbol{A}$
- Uma linha em $\boldsymbol{A}$
- Uma coluna em $\boldsymbol{A}$

conforme representado na Figura 4.

[221] Os componentes em cada conjunto $B_i$ podem ser computados aplicando-se uma função aleatorizada $\pi_i$ à $\boldsymbol{A_{master}}$, isto é, $B_i = \pi_i(\boldsymbol{A_{master}})$. Essa função aleatorizada pode ser uma permutação das entradas, caso o número de elementos na $\boldsymbol{A_{master}}$ e $B_i$ seja igual. No entanto, ela é suficiente caso obtenha os elementos em $A_{master}$ aleatoriamente.

[222] A função $\pi_i$ também pode obter alguns elementos em $\boldsymbol{A_{master}}$ e substituir alguns outros por elementos aleatórios. Por exemplo, metade das entradas do bloco $B_i$ são copiadas de valores da $A_{master}$ os outros são gerados aleatoriamente. Em ainda outra modalidade, a função $\pi_i$ obtém entradas exatamente suficientes da $\boldsymbol{A_{master}}$ e aplica uma função de substituição em cada uma das entradas, em que a função de substituição é uma função de um para um no grupo $\{0,1,..,q-1\}$. A função $\pi_i$ também pode permutar primeiramente as entradas e aplicar uma função de substituição nas entradas permutadas.

Sobre os parâmetros:

[223] Os elementos na matriz compartilhada $A$ podem estar em $[0,q-1]$. O módulo $q$ é cerca de $2^{14}$ ou $2^{16}$ em algumas modalidades. O tamanho de $\boldsymbol{A}$ é $k \times k$ com $k$ pode ser, por exemplo, 600 a 800. Um valor adequado para $N_B$ é cerca de $q$ de modo que todos os elementos sejam representados com alta probabilidade. Caso A seja construída a partir dos blocos, então, cada bloco pode ter o tamanho $k' \times k'$ com $k' < \sqrt{N_B}$, por exemplo, $k'$ pode ser de até 128 para $q = 2^{14}$.

Modalidades específicas:

[224] Em uma modalidade, é proposta uma permutação baseada em linha na qual A é computada em termos de

linha aplicando-se uma permutação total em $A$ que aplica uma permutação diferente em cada linha na $A_{master}$.

[225] Em uma modalidade, computa-se cada linha $\boldsymbol{a_i}$ em $A$ com $0 \leq i \leq k-1$ obtendo-se elementos da $A_{master}$, que a seguir $\boldsymbol{a_{master}}$ é denotado (definido como uma linha que contém $N_B$ elementos). Duas modalidades específicas são descritas:

- **Modalidade n°1:** Em uma primeira modalidade, a linha é computada como uma permutação $k$ de $N_b$. Por exemplo, $\boldsymbol{\pi_i}: \boldsymbol{a_i} = \boldsymbol{a_{master}}[r_i: (r_i + k - 1)\%N_b]$ em que $r_i$ é um valor aleatório. Isso leva a um total de

$$k^k$$

saídas aleatorizadas. A notação [a:b] denota todos os elementos com índices de a a b. Caso b seja menor que a, são obtidas todas as entradas de a até o fim do grupo compartilhado, e todas as entradas do início do grupo, até a. A notação $m\%N_b$ denota o número inteiro $m$ em $\{0,1,\dots,N_b-1\}$ que é diferente do número inteiro múltiplo de $N_b$ a partir de $m$.

- **Modalidade n° 2:** Em outro exemplo, $\boldsymbol{\pi_i}: \boldsymbol{a_{i\,[j]}} = \boldsymbol{a_{master}}\big[\big(\boldsymbol{r_i} + \boldsymbol{PRF}(\boldsymbol{s}|\boldsymbol{j})\big)\%\boldsymbol{N_b}\big]$, isto é, o $j^{ésimo}$ elemento em $a_i$ é obtido escolhendo-se um elemento em $\boldsymbol{a_{master}}$ aleatoriamente aplicando-se uma PRF que depende de uma semente e j, e deslocando-se os elementos um valor $r_i$ que depende da linha. Isso leva a muitas saídas aleatorizadas:

$$k^{k\binom{N_b}{k}}$$

no entanto, também exige a computação de uma PRF para cada linha, o que pode ser ineficiente. Os valores aleatórios $r_i$ são exemplos de parâmetros de seleção.

[226] A tabela a seguir resume o número de permutações e requisitos em número de bits computados durante a geração da $A_{master}$.

| Modalidade | n° de saídas aleatorizadas | n° de bits computados por meio de uma PRF |
|---|---|---|
| n° 1 | $k^k$ | $N_b \log q + k \log k$ |
| n° 2 | $k^{k\binom{N_b}{k}}$ | $N_b \log q + k(\log q + \log k)$ |

[227] Da tabela acima, pode-se observar que o número de saídas aleatorizadas é grande, mas o número de bits que precisa ser computado por meio de uma PRF é muito menor.

Uso em um protocolo:

[228] Diversos parâmetros precisam ser trocados em um protocolo. Por exemplo, um ou mais dentre:

- a semente usada para computar $A_{master}$
- a semente usada para computar a função $\pi_i$
- a especificação da função (permutação) $\pi_i$

[229] O método para atualizar a matriz compartilhada A pode ser usado em protocolos criptográficos, tais como trocas de chave, métodos de encapsulação de chave, sistemas de criptografia de chave pública, esquemas de assinatura etc. Deve-se notar que algumas modalidades também aprimoram o desempenho especialmente no lado do cliente (iniciador) em um KEX, visto que o iniciador tem de computar/armazenar menos. Isso também aprimora o lado do servidor (respondente), mas esse aprimoramento é relativamente menor.

[230] Quando aplicado a outros esquemas baseados em reticulado, esse método pode aprimorar o desempenho de esquemas de criptografia de chave pública ou esquemas de

assinatura, visto que o parâmetro público A também pode ser computado mais rapidamente quando recebe algumas sementes.

[231] Deve-se notar que isso poderia ser aplicado a módulo-reticulado, se cada módulo tivesse um parâmetro público diferente $a_i$ em que $a_i$ representa, neste caso, um polinômio em um determinado anel. Cada um dentre os polinômios públicos são obtidos aplicando-se uma permutação.

[232] A Figura 1 mostra esquematicamente um exemplo de uma modalidade de uma rede de concordância de chaves 100. A concordância de chaves é um exemplo de operação criptográfica. A concordância de chaves usa vários elementos, por exemplo, construir uma matriz compartilhada, por exemplo, uma matriz de chave privada, uma matriz de chave pública, uma multiplicação de matrizes, adição de ruído etc., que são comuns a outros tipos de operações criptográficas, por exemplo, criptografia de chave pública. Em uma modalidade, o dispositivo 100 pode ser reconfigurado para outros tipos de operações criptográficas, por exemplo, conforme indicado acima ou conforme indicado nas referências. Em uma modalidade, o dispositivo 100 pode ser reconfigurado para múltiplos tipos de operações criptográficas. De todo modo, o dispositivo 100 tem a vantagem de que vários níveis diferentes de estrutura e de dificuldade estão disponíveis. Por exemplo, a operação criptográfica pode ser qualquer uma dentre: um protocolo de troca de chaves (KEX), um método de encapsulação de chave (KEM), criptografia de chave pública (PKE), uma assinatura digital. Em uma modalidade, o primeiro nó de rede pode ser configurado para receber um seletor, selecionar uma operação criptográfica de múltiplas operações criptográficas diferentes, No entanto,

como exemplos, as modalidades a seguir supõe que o dispositivo 100 é configurado para concordância de chaves.

[233] São mostrados na Figura 1 dois nós de rede no sistema: um nó de rede 110 do tipo iniciador e um nó de rede 210 do tipo respondente. Em uma modalidade do sistema de concordância de chaves, o número de nós pode ser maior, até mesmo muito maior, que dois, por exemplo, mais de 1.000 nós, por exemplo, mais de 10^6 nós.

[234] A diferença entre um nó de rede do tipo iniciador ou respondente é como ele lida com os dados de reconciliação. Um nó de rede do tipo iniciador recebe dados de reconciliação e aplica-os para obter a chave compartilhada, enquanto um nó de rede do tipo respondente gera dados de reconciliação e envia-os ao nó de rede do tipo iniciador. O nó de rede do tipo respondente não precisa dos dados de reconciliação para obter a chave compartilhada. Tipicamente, um tipo iniciador também iniciará o protocolo de concordância de chave entre os dois nós de rede, visto que isso pode reduzir o número de rodadas executadas entre os dois nós de rede. Entretanto, isso não é necessário; o protocolo de concordância de chaves pode também ser iniciado por um nó de rede do tipo respondente.

[235] Além disso, em uma modalidade do nó de rede, o nó de rede é configurado para operar de acordo com um modo de iniciador e de acordo com um modo de respondente. Por exemplo, se o nó de rede inicia uma concordância de chave, por exemplo, envia uma mensagem para um outro nó de rede sinalizando o início do protocolo de concordância de chaves, então, o nó de rede pode mudar para modo de iniciador. Se o nó de rede responde a uma concordância de chave, por exemplo,

recebe uma mensagem a partir de um outro nó de rede sinalizando o início do protocolo de concordância de chaves, então, o nó de rede pode mudar para modo de respondente. Embora isso seja conveniente na prática, essa opção também não é estritamente necessária; por exemplo, em um sistema de concordância de chaves, alguns modos podem ser configurados apenas como iniciadores e alguns podem ser configurados apenas como nós respondentes. Uma consequência disso será que alguns nós não podem concordar sobre uma chave compartilhada em conjunto. Para algumas redes, isso não precisa ser um problema, por exemplo, em rede ad hoc, ou redes de distribuição sem fio ad hoc, etc., desde que muitos pares de nós de rede possam se comunicar e concordar sobre uma chave compartilhada.

[236] O nó de iniciador 110 compreende uma interface de comunicação 120. O nó de respondente 210 compreende uma interface de comunicação 220. As interfaces de comunicação podem ser dispostas para comunicação digital com os outros nós do sistema de concordância de chaves. Contudo, não é necessário que todos os nós no sistema possam ser alcançados em todos os momentos.

[237] As interfaces de comunicação 120 e 220 são dispostas para comunicação digital. Por exemplo, as interfaces de comunicação podem ser dispostas para se comunicarem através de uma rede de computadores. Por exemplo, a interface de comunicação pode ser disposta para comunicação sem fio, por exemplo, Wi-Fi, ZigBee, Bluetooth e similares, e/ou comunicação com fio, por exemplo, Ethernet, USB e similares. A comunicação entre os nós 110 e 210 pode também ser uma combinação de conexões com fio e sem fio. Por exemplo, os nós no sistema 100, incluindo

os nós 110 e 120, podem compreender um armazenamento eletrônico que contém um identificador de comunicação, que identifica de modo inequívoco o nó no sistema 100. Por exemplo, o identificador de comunicação pode ser incluído nas mensagens digitais trocadas entre os nós 110 e 210, por exemplo, para endereçar a mensagem. Por exemplo, o identificador de comunicação pode ser um endereço IP, um endereço MAC e similares.

[238] O nó de rede eletrônica é configurado para um protocolo de troca de chaves (KEX). O protocolo envolve trocar mensagens entre os nós 110 e 210 através das interfaces de comunicação 120 e 220 e realizar cálculos em, por exemplo, dados recebidos a partir do outro nó. A execução do protocolo de concordância de chaves é implementada em um circuito processador, cujos exemplos são mostrados abaixo. A Figura 1 mostra unidades funcionais que podem ser unidades funcionais do circuito processador. Por exemplo, a Figura 1 pode ser usada como um modelo de uma possível organização funcional do circuito processador. O circuito processador não é mostrado separado das unidades na Figura 1. Por exemplo, as unidades funcionais mostradas na Figura 1 podem também ser total ou parcialmente implementadas em instruções de computador que são armazenadas nos nós de rede e são executáveis por um microprocessador do nó de rede.

[239] O nó de iniciador 110 e o nó de respondente 210 são configurados para um protocolo de troca de chaves (KEX). Esquemas KEX envolvem a troca de dados públicos, frequentemente chamados de chaves públicas, por cada uma das partes, que são, então, independentemente usados pela outra parte juntamente com seus dados privados, frequentemente chamados de chave secreta, para calcular o segredo comum compartilhado. Um recurso

interessante de algumas modalidades é que o real segredo final compartilhado nunca é trocado entre as partes, nem mesmo na forma criptografada, mas é calculado independentemente pelas duas partes em cada extremidade. Isso resulta em um recurso desejável conhecido como sigilo futuro, que garante que até mesmo o comprometimento das chaves secretas de uma das partes a longo prazo por um invasor no futuro não comprometeria o sigilo de mensagens criptografadas trocadas no passado.

[240] As modalidades da invenção não dependem de uma terceira parte confiável para fornecer comunicação confidencial. O canal de comunicação entre as interfaces de comunicação 120 e 220 não precisa ser necessariamente um canal seguro. Invasores podem ser capazes de interceptar o canal de comunicação. Mesmo assim, a chave que é concordada entre os nós 110 e 210 pode ser segura. Se o canal de comunicação estiver protegido contra alterações, um grau de autenticação pode ser obtido na medida em que é fornecido pelo canal. No entanto, se o canal entre as interfaces de comunicação 120 e 220 não estiver protegido contra alterações, nenhum esquema KEX obterá a autenticação. Para obter a autenticação, as modalidades podem ser combinadas com qualquer mecanismo de autenticação conhecido, por exemplo, um mecanismo de autenticação implícita, por exemplo, com o uso de chaves públicas certificadas, ou um mecanismo de autenticação explícita, por exemplo, com o uso de assinaturas digitais.

[241] O nó de iniciador 110 compreende uma unidade de matriz compartilhada 130. O nó de respondente 210 compreende uma unidade de matriz compartilhada 230. As unidades de matriz compartilhada 130 e 230 são configuradas para obter uma matriz compartilhada (A) que é compartilhada

entre os dois nós. Há muitas maneiras de assegurar que a mesma matriz seja compartilhada entre os nós 110 e 210, especialmente devido ao fato de que a matriz A não precisa necessariamente ser mantida privada para os nós 110 e 210.

[242] A estrutura da matriz compartilhada *A* pode ser definida com um parâmetro de dificuldade *d*, e um parâmetro de estrutura *n*. A unidade de matriz compartilhada 130 e a unidade de matriz compartilhada 230 podem ser configuradas para receber como entrada o parâmetro de dificuldade *d*, e o parâmetro de estrutura *n*. Embora exista a vantagem de que o dispositivo pode suportar muitos tipos de problemas criptográficos subjacentes, isso não é necessário. A estrutura de *A*, por exemplo, seu tamanho e seu grau de polinômios (se houver), também pode ser determinada.

[243] Note-se que ambas as unidades usam a matriz compartilhada com o mesmo parâmetro de dificuldade *d* e parâmetro de estrutura *n*. Esses números *d* e *n* determinam o tamanho e o tipo de elementos de matriz *A*. Por exemplo, o parâmetro de dificuldade *d* e o parâmetro de estrutura *n* podem ser definidos através de uma API, por exemplo, por uma aplicação que usa o dispositivo 110 e 210, por exemplo, para comunicação protegida. A aplicação pode tomar uma decisão sobre a dificuldade e estrutura e instruir o dispositivo 110 e/ou 250, por exemplo, por uma chamada de função.

[244] As entradas na matriz compartilhada *A* são selecionadas módulo um primeiro módulo *q*, módulo um polinômio de redução (*f*) de grau igual ao parâmetro de estrutura (*n*). Se n=1, as entradas são números inteiros; se $n > 1$, elas são polinômios. O primeiro módulo *q* e o polinômio de redução *f* também são compartilhados entre os nós 110 e o nó 210, por exemplo,

comunicados ou predeterminados. A matriz compartilhada *A* é tipicamente uma matriz quadrada $k \times k$, por exemplo, de dimensão *k*. No entanto, não é necessário que *A* seja quadrada. Em vez disso, *A* pode ser retangular. Os tamanhos das matrizes de chave privada são, então, escolhidos para serem compatíveis com os tamanhos de *A*.

[245] O número de linhas e colunas *k* é igual ao parâmetro de dificuldade *d* dividido pelo parâmetro de estrutura *n*. Se $n = d$, a matriz *A* tem uma única entrada de polinômio. Uma escolha prática para o polinômio de redução *f* é, por exemplo, $x^n + 1$ ou $x^n - 1$.

[246] É preferencial que o parâmetro de estrutura *n* divida o parâmetro de dificuldade *d*, de modo que a dimensão *k* possa ser computada. Entretanto, em uma modalidade, *k* é obtido arredondando-se o parâmetro de dificuldade *d* dividido pelo parâmetro de estrutura *n*, por exemplo, arredondando-se para o número inteiro mais próximo, arredondando-se para cima ou arredondando-se para baixo.

[247] Em uma modalidade, o parâmetro de estrutura *n* está limitado para ser um número primo, mas isso não é necessário. Em uma modalidade, o parâmetro de estrutura *n* é uma potência de 2, e o primeiro módulo *q* é um número primo, mas isso também não é necessário.

[248] Os nós 110 e 210 têm um grupo compartilhado 131. O grupo compartilhado é compartilhado com o segundo nó de rede e que compreende múltiplos números inteiros e/ou polinômios para uso em uma entrada da matriz compartilhada. O grupo compartilhado pode, então, ser visualizado como um vetor de números inteiros e/ou polinômios. Por exemplo, podem ser fabricados nós com acesso a um grupo compartilhado. No entanto,

um modo preferencial de obter um grupo compartilhado é gerar um grupo compartilhado. Isso tem a vantagem de que diferentes pares de nós podem ter diferentes grupos compartilhados. Por exemplo, os nós 110 e 210 podem ser obtidos para gerar o grupo compartilhado por um gerador de números pseudoaleatórios determinísticos. O gerador de número aleatórios é semeado a partir de uma semente compartilhada. Por exemplo, um dos dispositivos, por exemplo, o dispositivo 110 ou o dispositivo 210, pode gerar uma semente e enviá-la à outra parte. Por exemplo, a primeira parte a enviar uma matriz de chave pública pode enviar uma semente para gerar o grupo compartilhado com ela, e a partir disso a matriz compartilhada. Por exemplo, as duas podem gerar uma semente e enviá-la para a outra parte. A semente usada para geração é obtida combinando-se ambas as sementes, por exemplo, adicionando-as.

[249] Quando o grupo compartilhado é gerado, ele pode ser retido, por exemplo, temporariamente em uma memória eletrônica, ou em um armazenamento de maior duração, por exemplo, em um dispositivo de gravação magnética, tal como um disco rígido ou uma memória não volátil. Manter o grupo compartilhado significa que ele não precisa ser gerado novamente. Em uma modalidade, o grupo compartilhado permanece o mesmo por múltiplas trocas de chave. No entanto, o grupo compartilhado também pode estar envolvido de algum modo. Por exemplo, em cada geração de chave nova, um número fixo de entradas no grupo pode ser descartado, e um novo grupo de números gerado, por exemplo, também formam uma semente, por exemplo, uma semente nova. Isso mantém a sobrecarga sob controle, mas garante que a longo prazo, o grupo compartilhado e, portanto, as entradas na matriz

compartilhada, serão renovados. Por exemplo, o número pode ser 10 etc.

[250] Entradas na matriz compartilhada ou coeficientes das entradas são selecionados a partir do grupo compartilhado. Há algumas opções. A matriz compartilhada pode compreender números inteiros, nesse caso o grupo compartilhado também compreende números inteiros. A matriz compartilhada pode compreender polinômios, nesse caso, o polinômio compartilhado compreende números inteiros, que podem ser selecionados como coeficientes para os polinômios na matriz compartilhada, ou o grupo compartilhado pode compreender polinômios, caso em que as entradas no grupo compartilhado podem ser usadas diretamente para entradas na matriz compartilhada. Nessas modalidades, as entradas no grupo compartilhado são, de preferência, módulo o primeiro módulo $q$, embora isso não seja estritamente necessário. De igual modo, caso as entradas no grupo compartilhado sejam polinômios, elas são, de preferência, módulo o módulo compartilhado $q$ e o polinômio de redução compartilhado $f$. Para a maioria das aplicações, não é necessário que as sementes, o grupo compartilhado, a matriz compartilhada, os parâmetros de seleção etc., sejam secretos, embora possam ser públicos.

[251] O tamanho do grupo compartilhado pode ser menor que o tamanho da matriz compartilhada. Por exemplo, o tamanho do armazenamento da matriz compartilhada pode ser proporcional ao número de entradas no tempo da matriz compartilhada, possivelmente vezes o grau do polinômio de redução menos 1. O tamanho do grupo compartilhado pode ser proporcional ao número de entradas dos mesmos, possivelmente vezes o grau do polinômio de redução menos 1. Em uma modalidade, o tamanho do grupo compartilhado é menor que o

tamanho da matriz compartilhada, por exemplo, menor ou igual à metade do tamanho da matriz compartilhada. Em uma modalidade, o tamanho do grupo compartilhado é menor que o tamanho da matriz compartilhada, por exemplo, menor que ou igual à raiz quadrada do tamanho da matriz compartilhada; isso tem a vantagem de que a matriz compartilhada pode ser dividida em $k$ partes que são iguais ou aproximadamente iguais ao tamanho do grupo compartilhado. Em uma modalidade, o número de entradas no grupo compartilhado é menor que o primeiro módulo $q$, ou um múltiplo pequeno do mesmo, por exemplo, $2q$, $3q$ etc. Uma vantagem do último é que praticamente todos os elementos módulo $q$ estão propensos a estar presentes no grupo compartilhado e, então, serão representados de modo aproximado na matriz compartilhada.

[252] Em uma modalidade, um grupo compartilhado tem tamanho maior que a matriz compartilhada, mas o nó usa o mesmo grupo compartilhado para múltiplas matrizes compartilhadas. De fato, um grupo compartilhado pode ser usado também com segundos nós diferentes.

[253] Uma vantagem de um grupo compartilhado é o fato de ter um baixo custo para atualizar a matriz compartilhada. Por exemplo, meros parâmetros de seleção podem ser enviados ou trocados, com base nos quais, uma nova matriz compartilhada pode ser selecionada rapidamente a partir do grupo compartilhado. Em particular, em algumas modalidades, são necessárias poucas ou nenhuma função.

[254] Em uma modalidade, os nós obtêm um ou mais parâmetros de seleção que são compartilhados com o outro nó de rede. Os parâmetros de seleção podem ser predeterminados, por exemplo, os nós podem ter acesso a uma sequência de parâmetros

de seleção e usar o próximo ou os próximos para gerar a matriz compartilhada. Tais parâmetros de seleção também podem ser selecionados por um nó e comunicados ao outro. Os parâmetros de seleção também podem ser selecionados por ambos os nós e comunicados ao outro, podendo então ser combinados, por exemplo, adicionados, por exemplo, submetidos a XOR.

[255] Os parâmetros de seleção são usados para controlar as funções de seleção, que obtêm, por exemplo, selecionam, as entradas na matriz compartilhada a partir do grupo compartilhado.

[256] Por exemplo, as unidades de matriz compartilhada 130 e 230 podem ser configuradas com uma ou mais funções de seleção. As funções de seleção atribuem um número inteiro, polinômio e/ou coeficiente de polinômio a partir do grupo compartilhado a uma entrada na matriz compartilhada, ao número inteiro específico, ao polinômio ou ao coeficiente de polinômio atribuído por uma função de seleção que depende do parâmetro de seleção.

[257] Por exemplo, a função de seleção pode ser funções $g_{i(r_i,j)}$ que tomam como entrada um parâmetro de seleção $r_i$ e um índice $j$ no grupo compartilhado ou parte do mesmo. A saída da função de seleção pode ser um par $(x,y)$ que indica uma entrada na matriz compartilhada A. A saída da função de seleção pode ser um par $(x,y,t)$ que indica uma entrada $(x,y)$ na matriz compartilhada e um grau $t$ de um coeficiente de um polinômio na entrada. Nesse caso, a função $g_{i(r_i,j)}$ é avaliada quanto a diferentes valores do índice $j$ para, por exemplo, o índice de $j = 0,\dots,N$, e a entrada no grupo compartilhado para o índice $j$ é atribuída à saída da função de seleção.

[258] Por exemplo, a função de seleção pode ser funções $g_{i(r_i,x,y)}$ ou $g_{i(r_i,x,y,t)}$ que tomam como entrada um parâmetro de seleção $r_i$ e um par $(x,y)$ que indica uma entrada na matriz compartilhada ou um triplo $(x,y,t)$ que indica uma entrada $(x,y)$ na matriz compartilhada e um grau $t$ de um coeficiente de um polinômio na entrada. A saída da função de seleção pode ser um índice $j$ no grupo compartilhado ou parte dele. Nesse caso, a função $g_{i(r_i,j)}$ é avaliada quanto a diferentes entradas na matriz compartilhada e/ou quanto a diferentes valores para t, e a entrada no grupo compartilhado é atribuída à entrada na matriz compartilhada. Um grupo compartilhado pode ser representado como a própria matriz A, nesse caso, o índice j pode ser representado como um par de índices.

[259] Por exemplo, a função de seleção pode ser permutação. A permutação é aplicada ao grupo compartilhado ou a uma parte do mesmo, e o resultado é atribuído de acordo com uma sequência predeterminada, por exemplo, linear, às entradas na matriz compartilhada, por exemplo, números inteiros, polinômios ou coeficientes.

[260] Em uma modalidade, a matriz compartilhada A é particionada em conjuntos. A cada conjunto uma função de seleção mapeia o grupo compartilhado ou parte do mesmo para o grupo, dependendo, pelo menos parcialmente, do parâmetro de seleção. Os exemplos de particionamento são mostrados na Figura 4. Na Figura 4 são mostradas quatro matrizes A: 410, 420, 430 e 440.

[261] Na matriz compartilhada 410, os conjuntos são blocos de matriz compartilhada A. Por exemplo, um bloco é pelo menos 2x2 ou maior. Um bloco pode, mas não precisa, ser quadrado. Um conjunto é indicado com 412.

[262] Na matriz compartilhada 420, os conjuntos são linhas de matriz compartilhada A. Um conjunto é indicado com 422.

[263] Na matriz compartilhada 430, os conjuntos são linhas de matriz compartilhada A. Um conjunto é indicado com 432.

[264] Na matriz compartilhada 440, os conjuntos são uma seleção aleatório de entradas da matriz compartilhada A. Um conjunto é indicado com 442. Por exemplo, também nesse caso, o conjunto pode ter um tamanho igual a $1/k$ vezes o número de entradas em A. Por exemplo, o conjunto 442 pode compreender k entradas, por exemplo, números inteiros. O conjunto 442 pode ser até mesmo uma seleção de $1/k$ vezes o número de coeficientes em A. A atribuição de coeficientes aos conjuntos, por exemplo, o conjunto 442, pode ser aleatória. Um coeficiente de uma entrada de polinômio em $A$ pode ser distribuído de um ou por mais de um conjunto.

[265] Um parâmetro de seleção $r_i$ pode ser, por exemplo, selecionar, um índice aleatório no grupo compartilhado. Em seguida, a sequência consecutiva de números inteiros no grupo compartilhado pode ser atribuída às entradas às quais a função $g_i$ é atribuída. Uma função de seleção pode indicar que uma dada permutação predeterminada deve ser aplicada primeiramente esse número de vezes. Essas vezes podem ser combinadas. Por exemplo, uma permutação aleatória de todo o grupo compartilhado pode ser fixa nos nós. Um primeiro parâmetro de seleção $r_0$ indica que a permutação fixa deve ser executada $r_0$ vezes, após isso $r_i$ pode ser usado para a função $g_i$, por exemplo, como um índice.

[266] Um parâmetro de seleção pode ser uma entrada para um gerador congruente linear, sendo que o gerador congruente linear atribuiu índices no grupo compartilhado aos índices em um conjunto da matriz compartilhada. Por exemplo, o parâmetro de seleção pode ser um multiplicador na relação de recorrência do multiplicador na relação de recorrência do gerador congruente linear. De preferência, o parâmetro de seleção é relativamente primo com o número de entradas em um conjunto. Por exemplo, um parâmetro de seleção também pode ser um incremento. O gerador congruente linear pode gerar tantos números quanto houver no conjunto e pode permutar todos ou parte do grupo compartilhado de acordo com o mesmo. Note-se que a sobrecarga de um gerador congruente linear é muito menor que a de uma função aleatória criptográfica.

[267] Uma função de seleção pode compreender também uma função aleatória criptográfica, por exemplo, com base nos parâmetros de seleção $r$ e $s$ as entradas no grupo compartilhado com índice $\left(r + Prf(s|j)\right)$ módulo o tamanho do grupo compartilhado podem ser mapeadas para uma entrada com índice $j$ de um conjunto de A, por exemplo, em uma linha ou coluna da matriz compartilhada, em que $Prf$ denota uma função pseudoaleatória, e $(s|j)$ denota a concatenação de $s$ e $j$.

[268] Em ainda outra modalidade, após a aplicação da função de seleção ao grupo, é aplicada uma função adicional S. Os exemplos de tal função S são a aplicação em termos de elementos de uma função um para um em $\{0,1,\dots,q-1\}$, ou a substituição de algumas entradas de matriz por entradas aleatorizadas.

[269] Em uma modalidade, as unidades de matriz compartilhada 130 e 230 são configuradas para selecionar todos

os números inteiros, polinômios e/ou coeficientes de polinômio na matriz compartilhada do grupo compartilhado. Isso tem a vantagem de que uma nova versão da matriz compartilhada torna-se rapidamente disponível. No entanto, qualquer tendência que esteja presente no grupo compartilhado pode ser refletida na matriz compartilhada. Para uma segurança muito alta, pode ser desejado evitar isso. Existem várias maneiras de se conseguir isso. Por exemplo, o conteúdo do grupo compartilhado pode ser atualizado ao longo do tempo. Por exemplo, após cada nova matriz compartilhada que é gerada, o conteúdo do grupo compartilhado pode ser atualizado, por exemplo, gerando-se um número de novas entradas com um gerador de número aleatório. No entanto, isso pode gerar o risco de o grupo compartilhado poder perder sincronia entre o primeiro e o segundo nó. Esse risco é reduzido com o uso dos parâmetros de seleção como uma semente para atualizar o grupo compartilhado. Por exemplo, o grupo compartilhado pode ser atualizado antes da geração da matriz compartilhada. As mudanças são mantidas no grupo compartilhado apenas se ocorrer a geração de chave bem-sucedida; de outro modo, o grupo compartilhado é revertido para a situação anterior. Caso a geração de chave bem-sucedida não seja possível, o grupo compartilhado também pode ser atualizado por regeneração completa, por exemplo, de acordo com um protocolo de regeneração de grupo compartilhado que pode compreender a troca de sementes para a regeneração.

[270] Outras maneiras de reduzir a dependência em um grupo compartilhado particular é tomar apenas parte da matriz compartilhada do grupo compartilhado. Por exemplo, em uma modalidade, parte dos números inteiros, polinômios e/ou coeficientes de polinômio na matriz compartilhada é

selecionada do grupo compartilhado, e para gerar a parte restante da matriz compartilhada. Por exemplo, os nós podem trocar uma semente, por exemplo, de maneira semelhante àquela em que uma semente para gerar o grupo compartilhado é trocada, e/ou de maneira semelhante àquela em que o parâmetro de seleção é trocado. A semente é usada para gerar parte da matriz compartilhada. Por exemplo, as entradas predeterminadas na matriz compartilhada podem ser atualizadas. Por exemplo, os lugares predeterminados podem ter pelo menos uma entrada em cada linha e/ou em cada coluna. Em uma modalidade, os coeficientes ou todos os polinômios podem ser atualizados. Em uma modalidade, várias entradas/coeficientes são selecionadas aleatoriamente, por exemplo, selecionadas por um gerador de números aleatórios, semeadas com a semente ou um parâmetro de seleção etc.

[271] O número de entradas/coeficientes a serem substituídos ou gerados pode ser um número fixo, por exemplo, pelo menos 10 ou pelo menos 100 etc. O número pode ser proporcional à matriz compartilhada, por exemplo, pelo menos a raiz quadrada do número de entradas/coeficientes não é obtido do grupo compartilhado, mas gerado e/ou substituído.

[272] Em vez de tomar apenas parte das entradas da matriz compartilhada, pode-se selecionar toda a matriz compartilhada do grupo compartilhado primeiramente, mas em seguida substituir parte dela por outros valores. Por exemplo, uma etapa de substituição ou de aleatorização pode emendar a matriz compartilhada antes de ser usada para troca de chaves. Por exemplo, uma ou mais dentre as entradas pode ser selecionada, por exemplo, predeterminada ou selecionada aleatoriamente ou até mesmo por protocolo e substituída por uma entrada gerada

aleatoriamente, por exemplo, de acordo com a semente etc. Alternativamente, um mapeamento pode ser aplicado a algumas ou a todas as entradas na matriz compartilhada. Por exemplo, uma aplicação em termos de entrada de um mapeamento de um em {0,1,..,$q$-1} pode ser aplicada a uma, mais de uma ou todas as entradas. A função de aleatorização pode ser aplicada ao mesmo tempo que a substituição/recolocações, mas a função de aleatorização também pode ser aplicada a mais, por exemplo, a todas as entradas/coeficientes.

[273] A seguir, um protocolo de troca de chaves é explicado em detalhes. Nessa modalidade, presume-se que a matriz compartilhada compreende números inteiros módulo o primeiro módulo $q$. No entanto, em outras modalidades de um protocolo de troca de chaves, a matriz compartilhada e a chave privada compreendem polinômios módulo o módulo q e o primeiro polinômio de redução $f$. A matriz de chave pública pode ser módulo o segundo módulo $p$ e o polinômio de redução. Em vez de usar arredondamento, pode-se usar outras fontes de ruído. O nó de iniciador 110 compreende uma unidade de matriz de chave privada 140. O nó de respondente 210 compreende uma unidade de matriz de chave privada 240. A unidade de matriz de chave privada 140 é configurada para gerar uma matriz de chave privada $S_I$; A unidade de matriz de chave privada 240 é configurada para gerar uma matriz de chave privada $S_R$; As entradas nas matrizes de chave privada são polinômios módulo $f$. Se o grau de $f$ é 1, então, na prática, as entradas são números inteiros.

[274] De modo geral, a matriz de chave privada é escolhida aleatoriamente. Por exemplo, as entradas na matriz de chave privada $(S_I, S_R)$ podem ser escolhidas de uma

distribuição de probabilidade aleatório uniforme. A distribuição também pode ser não uniforme, por exemplo, em que a probabilidade de uma entrada igual a zero é maior que a probabilidade de uma entrada diferente de zero, por exemplo, pelo menos duas vezes maior.

[275] Várias restrições podem ser impostas na matriz de chave privada, por exemplo, para aprimorar a segurança ou reduzir o tamanho de dados da mesma. A matriz de chave privada $(S_I, S_R)$ pode ser escolhida de maneira uniformemente aleatória a partir de matrizes de chave privadas candidatas, por exemplo, dentro das restrições. Por exemplo, em uma modalidade, os coeficientes dos elementos na matriz de chave privada estão limitados em valor absoluto por um limite (s), por exemplo, em que o dito limite é 2 $(s = 2)$, ou em que o limite é 1 $(s = 1)$, o último é correspondente ao binário assinado. Por exemplo, as colunas e/ou linhas da matriz de chave privada $(S_I, S_R)$ têm um peso de Hamming fixo ou limitado $(h_s)$.

[276] Os inventores descobriram surpreendentemente que, a escolha do limite pequeno tem dupla vantagem: as multiplicações de matriz pela matriz de chave privada são mais rápidas, e a distância entre as chaves brutas calculada em cada lado é menor (ver abaixo). O último significa que menos dados de reconciliação são necessários e/ou a probabilidade de uma falha no protocolo porque os nós concordaram com uma chave diferente é menor.

[277] Em uma modalidade, o limite $s$ é 1 $(s = 1)$. Ou seja, as entradas de uma matriz de chave privada têm apenas coeficiente que são os valores -1, 0 e 1. Vamos nos referir a isso também como “binário assinado”.

[278] As dimensões das matrizes de chave privada são escolhidas de modo que elas possam ser multiplicadas pela matriz $A$. Por exemplo, se $A$ é $k$ por $k$, então, a matriz de chave privada do nó iniciador pode ser uma matriz de $k$ por $\bar{k}$. Os tamanhos de $\bar{k}$ e $k$ são escolhidos para que sejam grandes o suficiente a fim de se obter um número suficiente de bits na chave compartilhada e obter um nível de segurança suficientemente alto.

[279] Além de restringir o tamanho das entradas da matriz de chave privada, outras vantagens são obtidas pela restrição do número de elementos diferentes de zero. Será feita referência ao peso de Hamming de uma coluna ou fileira de uma matriz como seu número de entradas diferentes de zero

[280] Os inventores investigaram diferentes formas para restringir o peso de Hamming de matrizes de chave privada. É geralmente suficiente restringir o peso de Hamming para as colunas ou para as fileiras, dependendo se a matriz de chave privada é multiplicada à esquerda ou à direita pela matriz $A$. Por exemplo, caso a matriz de chave privada seja multiplicada à direita com a matriz $A$ (por exemplo, $AS$) é suficiente limitar o peso de Hamming nas colunas da matriz de chave privada.

[281] Por exemplo, um limite superior pode ser selecionado para o peso de Hamming das colunas e/ou fileiras de uma matriz de chave privada. O limite superior pode ser o mesmo para todas as colunas e/ou fileiras. Em uma modalidade, as colunas e/ou linhas da matriz de chave privada $(S_I, S_R)$ têm o mesmo peso de Hamming fixo.

[282] A segurança é aprimorada se, submetida às condições, uma matriz de chave privada for escolhida de forma

uniformemente aleatória a partir das matrizes de chave privada candidatas, isto é, a partir das matrizes que satisfazem aos requisitos escolhidos, por exemplo, como os limites sobre as entradas e os pesos de Hamming em colunas ou fileiras. Por exemplo, caso se deseje fazer cumprir a condição de que o peso de Hamming de cada coluna seja 50, então, é vantajoso selecionar uma matriz de chave privada a partir do conjunto de todas as matrizes com as dimensões corretas que têm peso de Hamming de 50 para cada coluna. Algoritmos eficientes para selecionar uma cadeia de bits uniformemente aleatórios de qualquer peso de Hamming desejado são conhecidos.

[283] Uma maneira diferente de limitar o peso de Hamming das matrizes de chave privadas é escolher as colunas e/ou linhas de uma matriz de chave privada $(S_I, S_R)$ de uma distribuição de probabilidade. Por exemplo, as entradas na matriz de chave privada $(S_I, S_R)$ podem ser escolhidas a partir de uma distribuição de probabilidade não uniforme, sendo que a probabilidade de uma entrada zero é maior do que a probabilidade de uma entrada diferente de zero. Em uma modalidade, a distribuição de probabilidade é escolhida de modo que forneça um peso de Hamming esperado predeterminado para as colunas e/ou fileiras. Por exemplo, para escolher uma coluna de comprimento *n* e peso de Hamming esperado $h_s$, pode-se selecionar cada entrada como diferente de zero com a probabilidade $hs/n$. A entrada diferente de zero pode ser escolhida como 1 ou -1, por exemplo, com igual probabilidade.

[284] O nó de iniciador 110 compreende uma unidade de matriz de chave pública 150. O nó de respondente 210 compreende uma matriz de chave pública 250. A unidade de

matriz de chave pública computa uma matriz de chave pública da matriz $A$ e uma matriz de chave privada $S$.

[285] Os termos público e privado pretendem transmitir que a chave compartilhada não pode ser obtida apenas com o conhecimento das informações públicas, ou não sem conhecimento de algumas informações privadas. Não há nenhuma exigência de que as informações públicas sejam ativamente compartilhadas. Por exemplo, o protocolo de concordância de chaves poderia ser executado ao longo de um canal seguro (presumível), que mantém as chaves públicas protegidas dos interceptadores. Nesse caso, o protocolo de concordância de chaves fornece uma camada adicional de segurança, caso a segurança do canal seja violada.

[286] A unidade de matriz de chave pública computa uma matriz de chave pública $P$ ($P_I$, $P_R$ para o iniciador e o respondente respectivamente) computando-se um produto de matriz entre a matriz compartilhada (A) e a matriz de chave privada ($S_I$ ou $S_R$ respectivamente) módulo o primeiro módulo (q) e $f$, obtendo-se um produto de matriz e adicionando-se ruído.

[287] Note-se que essa multiplicação de matrizes intermediárias não é revelada. O conhecimento da matriz compartilhada $A$ e o resultado dessa multiplicação de matrizes revelariam a chave privada, uma vez que ela pode ser calculada invertendo-se a matriz $A$.

[288] A adição de ruído pode ser feita selecionando-se termos de erro ou termos de ruído e adicionando-os à entrada do produto de matriz acima. Por exemplo, o ruído pode ser selecionado a partir de uma distribuição Gaussiana com média 0. Um modo preferencial para adicionar o ruído é diminuir

o produto de matrizes intermediárias, por exemplo, diminuir os coeficientes.

[289] Em uma modalidade, a unidade de matriz de chave pública diminui as entradas no produto de matriz para um segundo módulo $p$. O segundo módulo p é menor que o primeiro módulo $q$. Uma entrada escada é igual à entrada não escalada multiplicada pelo segundo módulo (p), dividido pelo primeiro módulo (q) e arredondado para o número inteiro mais próximo. Por exemplo, se $x$ é um módulo $q$ de entrada não escalada no produto de matriz, a entrada escalada pode ser selecionada como $\left\lfloor \frac{p}{q} x \right\rceil$, em que $\lfloor \rceil$ representa o número inteiro mais próximo. Após a operação de escalonamento, já não é diretamente possível calcular a chave privada a partir da chave pública e da matriz $A$.

[290] As entradas em uma matriz de chave pública podem ser representadas como polinômios com coeficientes no intervalo $[0, p-1]$. As entradas em uma matriz de chave privada também podem ser representadas com os coeficientes no intervalo $[0, p-1]$. Há uma vantagem em se escolher os coeficientes no intervalo $\left[-\frac{p}{2}, \frac{p}{2}\right]$ para reduzir o tamanho dos números inteiros para multiplicações subsequentes. Como destacado acima, o polinômio de chave privada pode também ter coeficientes escolhidos no intervalo $[-s, s]$ ou até mesmo $[0, s]$.

[291] A multiplicação da matriz $A$ e uma matriz de chave privada $S$ é feita módulo o primeiro módulo $q$ e $f$. Para essa finalidade, o nó de rede pode compreender uma unidade de redução modular para reduzir o módulo $q$ e o módulo $f$. Caso os coeficientes na matriz de chave privada $S$ sejam pequenos, por exemplo, limitados por 1, ou limitados por 1 em valor absoluto, a redução modular pode ser simplificada; durante a

multiplicação de matrizes, toda vez que um coeficiente se tornar maior que $q$ ou menor que 0 o resultado retorna para o intervalo 0 até $q-1$ subtraindo-se ou adicionando-se $q$.

[292] Tanto o nó de iniciador quanto o nó de respondente enviam sua matriz de chave pública para o outro nó, por exemplo, com o uso das interfaces de comunicação 120 e 220. Os inventores descobriram pelo menos três vantagens que são desenvolvidas escalonando-se para baixo a produto matricial. Em primeiro lugar, a geração e a adição explícitas de ruído no produto matricial são evitadas. A introdução de ruído exige o cálculo de uma distribuição de probabilidade, por exemplo, uma distribuição gaussiana. Esses cálculos são relativamente intensos. Segundo, os requisitos de comunicação são reduzidos. Visto que o segundo módulo $p$ é menor que o primeiro módulo $q$, são necessários para representar uma entrada da matriz de chave pública menos bits que de uma matriz de chave privada. Terceiro, cálculos que envolvem a matriz de chave pública são menores, visto que eles envolvem números menores. É surpreendente que uma única medida forneça simultaneamente três vantagens.

[293] É muito preferencial que o segundo módulo $p$ divide o primeiro módulo $q$. Curiosamente, os inventores descobriram que nem o primeiro nem o segundo módulo precisam ser um número primo. De fato, constata-se que a escolha do segundo módulo ($p$) e/ou do primeiro módulo ($q$) como uma potência de 2 tem uma vantagem de que as chaves públicas e chaves privadas são distribuídas uniformemente. Em uma modalidade, tanto o primeiro quanto o segundo módulo são uma potência de 2.

[294] Em uma modalidade, é necessário adicionalmente $2^{B+b_h}|p$ além dos módulos $p$ e $q$ que são potências

de dois. Isso faz com que as chaves compartilhadas sejam uniformes mesmo se os dados de reconciliação forem observados. *B* é o número de bits de chave compartilhada extraído por entrada de chave bruta.

[295] Os tamanhos dos módulos não precisam ser muito grandes. Por exemplo, em uma modalidade, o segundo módulo tem tamanho de bit de 12 ou mais, e/ou o primeiro módulo tem tamanho de bit de 8 ou mais. Tamanhos maiores ou menores são possíveis dependendo dos requisitos de segurança. Em uma modalidade, *q* está na faixa de 2^12 e 2^15, *p* está na faixa de 2^7 e 2^9 (inclusive). Os valores de *p* e *q* podem ser escolhidos maiores ou menores de acordo com as demandas de segurança.

[296] O nó de iniciador 110 compreende uma unidade de chave compartilhada 160. O nó de respondente 210 compreende uma unidade de chave compartilhada 260. As unidades de chave compartilhada são diferentes no sentido de que elas geram e transmitem, ou recebem e aplicam os dados de reconciliação.

[297] Tanto a unidade de chave compartilhada 160 quanto a unidade de chave compartilhada 260 são configuradas para calcular uma chave bruta 162, 262 como um produto matricial entre a chave pública recebida do outro nó e a matriz de chave privada do próprio nó de rede. O produto é computado módulo o polinômio de redução *f*. Caso o escalonamento seja usado, o último computado também módulo o segundo módulo (*p*). As dimensões das matrizes e as multiplicações de matriz são escolhidas de modo que, se a operação de escalonamento for omitida, ambas as partes calculariam uma chave bruta idêntica. Ou seja, chaves idênticas resultariam sem adicionar ruído, e todos os cálculos foram feitos módulo q e *f*. No entanto, devido ao ruído, as duas chaves brutas

não precisam ser idênticas. A computação da chave bruta pode ser feita módulo *p*. Os nós de rede podem compreender uma unidade modular para reduzir o resultado das multiplicações módulo *p*.

[298] A unidade de chave compartilhada 260 do nó de respondente 210 é disposta para obter a chave compartilhada 266 e os dados de reconciliação 264 a partir da chave bruta 262 e para enviar os dados de reconciliação 264 para o nó de rede de iniciador 110. Os dados de reconciliação podem assumir a forma de um ou mais bits na chave bruta, por exemplo, bits dos coeficientes. Bits escolhidos como dados de reconciliação são ignorados com o propósito de gerar uma chave. As referências contêm vários exemplos de dados de reconciliação.

[299] A unidade de chave compartilhada 260 escolhe alguns bits das entradas da chave bruta para formar uma chave. Por exemplo, os bits escolhidos podem ser concatenados. Em uma modalidade, os bits escolhidos são inseridos em uma função de derivação de chave (“KDF” - Key Derivation Function), por exemplo, uma função hash criptográfica. Um exemplo de uma KDF é dado, por exemplo, na CMLA_KDF da especificação técnica CMLA, versão: V1.43-20131218, ou a função KDF definida na “especificação DRM”, OMA-TS-DRM-DRM-V2_0_2-20080723-A, Open Mobile Alliance™, versão 2.0.2, seção 7.1.2, etc. A função de derivação de chave pode ser aplicada às entradas de bits de chave na chave bruta, por exemplo, obtidas pela função de arredondamento, por exemplo, após a concatenação, ou a partir das saídas da função de reconciliação, por exemplo, também após a concatenação.

[300] Alguns bits que não são escolhidos como bits de chave podem ser selecionados como dados de reconciliação. Finalmente, alguns bits podem ser

completamente descartados. Os dados de reconciliação de chave obtidos a partir da chave bruta por unidade de chave compartilhada 260 são enviados para o nó de iniciador 110.

[301] A unidade de chave compartilhada 160 é configurada para receber os dados de reconciliação 164 ($h$) do segundo nó de rede, e para calcular uma chave compartilhada aplicando-se uma função de reconciliação aos dados de reconciliação recebidos e à matriz de chave bruta 162. Por exemplo, a função de reconciliação pode ser aplicada a cada uma das entradas na chave bruta 162 e à parte correspondente dos dados de reconciliação. Por exemplo, se os dados de reconciliação 164 faziam parte da chave bruta gerada pela unidade de respondente 210, o nó de iniciador pode selecionar uma chave bruta que poderia ter sido obtida pelo nó 210 e é compatível com os dados de reconciliação recebidos, por exemplo, tem os mesmos bits intermediários conforme recebido.

[302] Pode-se reutilizar uma dentre a matriz $A$ e a matriz de chave privada para múltiplas execuções do protocolo de concordância de chave (se ambas forem iguais, seria obtida a mesma chave compartilhada). Isso reduziria a sobrecarga de comunicação, especialmente no caso de a matriz $A$ ser reutilizada. Entretanto, os inventores perceberam que não há necessidade de reutilização de qualquer uma dentre a matriz $A$ e a matriz de chave privada, visto que nenhuma autenticação está conectada a esses elementos. Em uma modalidade preferencial, uma nova matriz $A$ e uma nova chave privada são obtidas para cada nova troca de chaves. Isso tem a vantagem de que os invasores não têm a opção de observar informações adicionais ao observar múltiplas execuções do protocolo. Além disso, o sigilo direto é aprimorado.

[303] Depois de o protocolo ser concluído e ambos os nós terem calculado uma chave compartilhada, um dos nós pode enviar uma mensagem de confirmação de chave para o outro nó para verificar se eles concordaram com a mesma chave. Por exemplo, uma mensagem de confirmação de chave pode ser um hash da chave compartilhada, uma criptografia de um valor fixo, uma descriptografia de um valor aleatório juntamente com o valor aleatório. A confirmação de chave pode também ser realizada com o uso de um protocolo de resposta de desafio. Pode-se também escolher omitir a confirmação de chave. Se as partes obtiverem uma chave compartilhada diferente, então, a comunicação subsequente realizada entre elas falhará. Por exemplo, a chave compartilhada pode ser usada para criptografar e/ou autenticar outras comunicações, por exemplo, mensagens digitais. Se eles chegaram a uma chave compartilhada diferente, então, a descriptografia e/ou verificação podem falhar.

[304] Nas várias modalidades, a interface de comunicação pode ser selecionada dentre várias alternativas. Por exemplo, a interface de comunicação pode ser uma interface de rede para uma rede local ou de longa distância, por exemplo, a Internet, uma interface de armazenamento para um armazenamento de dados interno ou externo, um teclado, etc.

[305] Os nós de rede podem compreender um armazenamento eletrônico, por exemplo, para armazenar dados intermediários, como a matriz A, as matrizes de chave pública e privada e a chave compartilhada, etc. O armazenamento pode ser implementado como uma memória eletrônica, diga-se uma memória flash, ou memória magnética, diga-se disco rígido ou similares. O armazenamento pode compreender múltiplas memórias distintas que juntas compõem o armazenamento. O armazenamento também pode

ser uma memória temporária, diga-se uma memória RAM. No caso de um armazenamento temporário, o armazenamento pode usar alguns meios para obter parâmetros comuns antes do uso, por exemplo, obtendo-os em uma conexão de rede opcional (não mostrada separadamente).

[306] Tipicamente, os dispositivos 110 e 210 compreendem, cada um, um microprocessador (não mostrado separadamente na Figura 1) que executa o software adequado armazenado nos dispositivos 110 e 210; por exemplo, aquele software pode ter sido baixado e/ou armazenado em uma memória correspondente, por exemplo uma memória volátil, como RAM, ou uma memória não volátil, como Flash (não mostrada separadamente). Alternativamente, os dispositivos 110 e 210 podem ser total ou parcialmente implementados em lógica programável, por exemplo, como arranjo de portas programáveis em campo (FPGA - "field-programmable gate array"). Os dispositivos 110 e 210 podem ser implementados, total ou parcialmente, como um "circuito integrado para aplicação específica" (ASIC - "application-specific integrated circuit"), isto é, um circuito integrado (CI) personalizado para seu uso específico. Por exemplo, os circuitos podem ser implementados em CMOS, por exemplo, com o uso de uma linguagem de descrição de hardware como Verilog, VHDL etc.

[307] Em uma modalidade, um nó de rede compreende um circuito de interface de comunicação, um circuito de matriz compartilhada, um circuito de matriz de chave privada, um circuito de matriz de chave pública e um circuito de chave compartilhada. Os circuitos implementam as unidades correspondentes descritas no presente documento. Os circuitos podem ser um circuito processador e circuito de armazenamento,

sendo que o circuito processador executa instruções representadas eletronicamente nos circuitos de armazenamento.

[308] Um circuito processador pode ser implementado de maneira distribuída, por exemplo, como múltiplos circuitos subprocessadores. Um armazenamento pode ser distribuído em múltiplos subarmazenamentos distribuídos. Toda a memória ou parte da memória pode ser uma memória eletrônica, memória magnética, etc. Por exemplo, o armazenamento pode ter uma parte volátil e uma parte não volátil. Parte do armazenamento pode ser apenas de leitura. Os circuitos podem também ser FPGA, ASIC ou similares.

[309] A Figura 2 mostra esquematicamente um exemplo de uma modalidade de um método de troca de chave eletrônica. O método pode ser executado por um primeiro nó de rede eletrônica, como nó de iniciador 110 ou nó de respondente 210.

[310] O método 400 compreende

- dispor (410) comunicação digital entre o primeiro nó de rede e um segundo nó de rede,
- obter (420) uma matriz compartilhada ($A$), sendo que a matriz compartilhada é compartilhada com o segundo nó de rede através da interface de comunicação, sendo que as entradas na matriz compartilhada $A$ são números inteiros, de preferência, números inteiros módulo um primeiro módulo $q$, e/ou as entradas na matriz compartilhada ($A$) são e/ou polinômios, sendo que a obtenção da matriz compartilhada ($A$) compreende:
  - obter um ou mais parâmetros de seleção compartilhados com o segundo nó de rede,
  - selecionar números inteiros, polinômios e/ou coeficientes de polinômio a partir de um grupo compartilhado,

em que o grupo compartilhado é compartilhado com o segundo nó de rede e compreende múltiplos números inteiros e/ou polinômios para uso em uma entrada da matriz compartilhada, sendo que a seleção é feita de acordo com uma ou mais funções de seleção, em que uma função de seleção atribui um número inteiro, polinômio e/ou coeficiente de polinômio a partir do grupo compartilhado para uma entrada na matriz compartilhada, em que o número inteiro, polinômio ou coeficiente de polinômio particular atribuído por uma função de seleção é dependente do parâmetro de seleção,

- gerar (430) uma matriz de chave privada ($S_I$), sendo que as entradas na matriz de chave privada são módulos de números inteiros ($q$) e/ou polinômios de número inteiro módulo o polinômio de redução ($f$),

- gerar (440) uma matriz de chave pública ($P_I$) por meio de

- computar (442) um produto de matriz entre a matriz compartilhada ($A$) e a matriz de chave privada ($S_I$) módulo o primeiro módulo ($q$) e/ou módulo o polinômio de redução ($f$) obtendo-se um produto de matriz e adicionando-se ruído (444) às entradas no produto de matriz,

- enviar (452) a matriz de chave pública do primeiro nó de rede para o segundo nó de rede,

- receber (454) uma matriz de chave pública ($P_R$) do segundo nó de rede,

- computar (460) uma chave bruta como um produto de matriz entre a chave pública recebida do segundo nó e a matriz de chave privada do primeiro nó de rede módulo o segundo módulo ($p$),

[311] Se o primeiro nó de rede operar de acordo com um modo de iniciador, então, o primeiro nó de rede executa os seguintes elementos adicionais.

- receber (472) dados de reconciliação ($h$) do segundo nó de rede,
- calcular (482) uma chave compartilhada aplicando-se uma função de reconciliação ($rec$) aos dados de reconciliação recebidos e à chave bruta.

Se o primeiro nó de rede operar de acordo com um modo de respondente, então, o primeiro nó de rede executa os seguintes elementos adicionais.

- obter (474) a chave compartilhada e os dados de reconciliação a partir da chave bruta,
- enviar (484) os dados de reconciliação para o primeiro nó de rede.

[312] Muitas formas diferentes de execução do método são possíveis, conforme ficará evidente para o versado na técnica. Por exemplo, a ordem das etapas pode variar ou algumas etapas podem ser executadas em paralelo. Além disso, outras etapas de método podem ser inseridas entre as etapas. As etapas inseridas podem representar modificações no método, conforme aqui descrito, ou podem não estar relacionadas ao método. Por exemplo, uma dada etapa pode não ter sido concluída completamente antes de uma próxima etapa ser iniciada.

[313] Um método de acordo com a invenção pode ser executado com o uso de um software que compreende as instruções para fazer com que um sistema processador execute o método 400. O software pode incluir apenas aquelas etapas empregadas por uma subentidade específica do sistema. O software pode ser

armazenado em uma mídia de armazenamento adequada, como um disco rígido, um disquete, uma memória, um disco óptico etc. O software pode ser enviado como um sinal por uma rede com fio, ou sem fio, ou com o uso de uma rede de dados, por exemplo a internet. O software pode ser disponibilizado para download e/ou para uso remoto em um servidor. Um método de acordo com a invenção pode ser executado com o uso de um fluxo de bits disposto para configurar uma lógica programável, por exemplo uma matriz de portas programável em campo (FPGA), para executar o método.

[314] Deve-se entender que a invenção também se estende a programas de computador, especialmente programas de computador armazenados em uma portadora, adaptados para colocar a invenção em prática. O programa pode estar em formato de código fonte, código objeto, uma fonte de códigos intermediários, um código objeto em formato parcialmente compilado, ou em qualquer outro formato adequado para uso na implementação do método de acordo com a invenção. Uma modalidade relacionada a um produto de programa de computador compreende instruções executáveis por computador que correspondem a cada uma das etapas de processamento de pelo menos um dos métodos apresentados. Essas instruções podem ser subdivididas em sub-rotinas e/ou ser armazenadas em um ou mais arquivos que podem estar estática ou dinamicamente ligados. Outra modalidade relacionada a um produto de programa de computador compreende instruções executáveis por computador que correspondem a cada um dos meios de pelo menos um dos sistemas e/ou produtos apresentados.

[315] A Figura 3a mostra uma mídia legível por computador 1000 que tem uma parte gravável 1010 que compreende um programa de computador 1020, sendo que o programa de

computador 1020 compreende instruções para fazer com que o sistema processador execute uma operação criptográfica, por exemplo, um método de concordância de chave, de acordo com uma modalidade. O programa de computador 1020 pode ser incorporado na mídia legível por computador 1000 como marcadores físicos ou por meio de magnetização da mídia legível por computador 1000. Entretanto, qualquer outra modalidade adequada também é concebível. Além disso, deve-se considerar que, embora a mídia legível por computador 1000 seja mostrada aqui como um disco óptico, a mídia legível por computador 1000 pode ser qualquer mídia legível por computador adequada, como um disco rígido, memória de estado sólido, memória Flash etc., e pode ser gravável ou não gravável. O programa de computador 1020 compreende instruções para fazer com que um sistema de processador realize o dito método de concordância de chaves 400 ou outra operação criptográfica (ou operações criptográficas) de acordo com uma modalidade.

[316] A Figura 3b mostra uma representação esquemática de um sistema processador 1140 de acordo com uma modalidade, por exemplo, para implementar uma ou mais operações criptográficas. O sistema processador compreende um ou mais circuitos integrados 1110. A arquitetura dos um ou mais circuitos integrados 1110 é esquematicamente mostrada na Figura 3b. O circuito 1110 compreende uma unidade de processamento 1120, por exemplo uma CPU, para executar componentes de programas de computador para executar um método de acordo com uma modalidade e/ou implementar seus módulos ou unidades. O circuito 1110 compreende uma memória 1122 para armazenar códigos de programação, dados etc. Parte da memória 1122 pode ser apenas de leitura. O circuito 1110 pode compreender um

elemento de comunicação 1126, por exemplo, uma antena, conectores ou ambos e similares. O circuito 1110 pode compreender um circuito integrado dedicado 1124 para executar parte ou todo o processamento definido no método. O processador 1120, a memória 1122, o CI dedicado 1124 e o elemento de comunicação 1126 podem ser conectados entre si através de um interconector 1130, como um barramento. O sistema processador 1110 pode ser disposto para comunicação com contato e/ou sem contato, com o uso de uma antena e/ou conectores, respectivamente.

[317] Por exemplo, em uma modalidade, um nó de rede pode compreender um circuito processador e um circuito de memória, sendo que o processador é disposto de modo a executar um software armazenado no circuito de memória. Por exemplo, o circuito processador pode ser um processador Intel Core i7, um ARM Cortex-R8, etc. Em uma modalidade, o circuito processador pode ser um ARM Cortex M0. O circuito de memória pode ser um circuito ROM ou uma memória não volátil, por exemplo uma memória Flash. O circuito de memória pode ser uma memória volátil, por exemplo, uma memória SRAM. No último caso, o dispositivo de verificação pode compreender uma interface de software não volátil, por exemplo, um disco rígido, uma interface de rede etc., disposta de modo a fornecer o software.

[318] Deve-se notar que as modalidades mencionadas acima ilustram ao invés de limitarem a invenção, e que os versados na técnica têm a capacidade de projetar muitas modalidades alternativas.

[319] Nas reivindicações, quaisquer sinais de referência colocados entre parênteses não devem ser interpretados como limitadores da reivindicação. O uso do

verbo “compreender” e suas conjugações não exclui a presença de elementos ou etapas além daquelas mencionadas em uma reivindicação. O artigo indefinido “um” ou “uma” que precede um elemento não exclui a presença de uma pluralidade desses elementos. A invenção pode ser implementada por meio de hardware que compreende vários elementos distintos, e por meio de um computador programado adequadamente. Na reivindicação de dispositivo que enumera vários meios, vários desses meios podem ser incorporados por um único item de hardware. O simples fato de certas medidas serem mencionadas em reivindicações dependentes mutuamente diferentes não indica que uma combinação dessas medidas não possa ser usada com vantagem.

[320] Nas reivindicações, as referências entre parênteses se referem a sinais de referência nos desenhos de modalidades exemplificadoras ou a fórmulas de modalidades, aumentando, assim, a inteligibilidade da reivindicação. Essas referências não devem ser interpretadas como limitadoras da reivindicação.

REIVINDICAÇÕES

1. PRIMEIRO NÓ DE REDE ELETRÔNICA (110) CONFIGURADO PARA UMA OPERAÇÃO CRIPTOGRÁFICA, sendo o primeiro nó de rede caracterizado por compreender

- uma interface de comunicação (120) disposta para comunicação digital com um segundo nó de rede,

- um circuito de processador configurado para

- obter uma matriz compartilhada ($A$), em que a matriz compartilhada é compartilhada com o segundo nó de rede através da interface de comunicação, entradas na matriz compartilhada $A$ são números inteiros e/ou entradas na matriz compartilhada ($A$) são polinômios de número inteiro, obter a matriz compartilhada ($A$) compreende

- obter um ou mais parâmetros de seleção compartilhados com o segundo nó de rede,

- selecionar números inteiros, polinômios e/ou coeficientes de polinômio a partir de um grupo compartilhado, em que o grupo compartilhado é compartilhado com o segundo nó de rede e compreende múltiplos números inteiros e/ou polinômios para uso em uma entrada da matriz compartilhada, sendo que a seleção é feita de acordo com uma ou mais funções de seleção, em que uma função de seleção atribui um número inteiro, polinômio e/ou coeficiente de polinômio a partir do grupo compartilhado para uma entrada na matriz compartilhada, em que o número inteiro, polinômio ou coeficiente de polinômio particular atribuído por uma função de seleção é dependente do parâmetro de seleção,

- gerar uma matriz de chave privada ($S_I$), em que entradas na matriz de chave privada são números inteiros e/ou polinômios de número inteiro,

- gerar uma matriz de chave pública ($P_I$)

- computando-se um produto de matriz entre a matriz compartilhada ($A$) e a matriz de chave privada ($S_I$) se obtém um produto de matriz, e adicionando-se ruído ao produto de matriz, e

- enviar a matriz de chave pública do primeiro nó de rede ao segundo nó de rede.

2. PRIMEIRO NÓ DE REDE ELETRÔNICA (110), de acordo com a reivindicação 1, caracterizado pelas entradas na matriz compartilhada $A$ serem números inteiros módulo um primeiro módulo $q$ e/ou entradas na matriz compartilhada ($A$) serem polinômios de número inteiro módulo um primeiro módulo $q$ e módulo um polinômio de redução $f$, e em que as entradas na matriz de chave privada são números inteiros módulo o primeiro módulo ($q$) e/ou polinômios de número inteiro módulo o primeiro módulo ($q$) e módulo o polinômio de redução ($f$), e em que a matriz de chave privada é computada módulo o primeiro módulo ($q$) e/ou módulo o primeiro módulo ($q$) e o polinômio de redução ($f$).

3. PRIMEIRO NÓ DE REDE, de acordo com qualquer uma das reivindicações anteriores, caracterizado por estar disposto para obter uma matriz compartilhada diferente ($A$) para cada nova troca de chave.

4. PRIMEIRO NÓ DE REDE, de acordo com qualquer uma das reivindicações anteriores, caracterizado por

- a matriz compartilhada e o grupo compartilhado compreenderem números inteiros ou polinômios, e os polinômios de seleção selecionarem números inteiros ou polinômios a partir do grupo compartilhado, em que o grupo compartilhado compreende menos entradas que a matriz compartilhada, ou

- a matriz compartilhada compreender polinômios e o grupo compartilhado compreender números inteiros, e os polinômios de seleção selecionarem coeficientes de polinômio a partir do grupo compartilhado, sendo que o grupo compartilhado compreende menos entradas que o número de entradas na matriz compartilhada vezes o grau do polinômio de redução menos 1.

5. PRIMEIRO NÓ DE REDE, de acordo com qualquer uma das reivindicações anteriores, caracterizado por estar disposto para obter o grupo compartilhado ($A_master$) gerando-se uma semente aleatória e enviando-se a semente aleatória através do canal de comunicação para o outro nó, em que o primeiro e o segundo nós de rede usam a semente aleatória para semear um gerador de números pseudoaleatórios determinísticos, o que gera o grupo compartilhado a partir da saída do gerador de números pseudoaleatórios.

6. PRIMEIRO NÓ DE REDE, de acordo com qualquer uma das reivindicações anteriores, caracterizado por

- entradas na matriz compartilhada e entradas no grupo compartilhado serem números inteiros módulo um primeiro módulo ($q$)

- entradas na matriz compartilhada serem polinômios de número inteiro módulo um polinômio de redução ($f$) e entradas no grupo compartilhado serem números inteiros módulo o primeiro módulo ($q$),

- entradas na matriz compartilhada serem polinômios de número inteiro e entradas no grupo compartilhado serem polinômios módulo o primeiro módulo ($q$) e o polinômio de redução ($f$).

7. PRIMEIRO NÓ DE REDE, de acordo com qualquer uma das reivindicações anteriores, caracterizado pelas entradas

na matriz compartilhada A serem repartidas em conjuntos, para cada conjunto uma função de seleção mapeia pelo menos parte do grupo compartilhado para o conjunto, na dependência de pelo menos parte do parâmetro de seleção.

8. PRIMEIRO NÓ DE REDE, de acordo com a reivindicação 7, caracterizado por

- os conjuntos corresponderem às fileiras ou colunas da matriz compartilhada, ou

- os conjuntos corresponderem à matriz compartilhada de blocos, e/ou

- em que os conjuntos têm tamanho igual.

9. PRIMEIRO NÓ DE REDE, de acordo com qualquer uma das reivindicações anteriores, caracterizado por uma função de seleção tomar como entrada um parâmetro de seleção e um índice que indica uma entrada ou coeficiente de uma entrada da matriz compartilhada, e produzir como saída um índice no grupo compartilhado.

10. PRIMEIRO NÓ DE REDE, de acordo com qualquer uma das reivindicações anteriores, caracterizado por uma função de seleção mapear uma sequência consecutiva de entradas do grupo compartilhado para uma sequência consecutiva de elementos em uma fileira ou coluna da matriz compartilhada.

11. PRIMEIRO NÓ DE REDE, de acordo com qualquer uma das reivindicações anteriores, caracterizado por

- uma função de seleção dependente de um parâmetro de seleção $r$ mapear entradas no grupo compartilhado com índice $r$ até $(r+k-1)$ módulo o tamanho do grupo compartilhado para uma fileira ou coluna da matriz compartilhada, em que $k$ é a dimensão da matriz, ou

- uma função de seleção dependente de parâmetros de seleção $r$ e $s$ mapea entradas no grupo compartilhado com índice $(r + Prf(s,j))$ módulo o tamanho do grupo compartilhado para uma entrada com índice $j$ de uma fileira ou coluna da matriz compartilhada, em que $Prf$ denota uma função pseudoaleatória que assume $s$ como uma semente e índice $j$ como entrada.

12. PRIMEIRO NÓ DE REDE ELETRÔNICA (110), de acordo com qualquer uma das reivindicações anteriores, caracterizado pela operação criptográfica ser um protocolo de troca de chave (KEX), em que o circuito de processador é configurado para

- receber uma matriz de chave pública ($P_R$) do segundo nó de rede,

- computar uma chave bruta como um produto de matriz entre a chave pública recebida do segundo nó e a matriz de chave privada do primeiro nó de rede,

em que o primeiro nó de rede é configurado adicionalmente para

- receber dados de reconciliação ($h$) do segundo nó de rede,

- computar uma chave compartilhada aplicando-se uma função de reconciliação ($rec$) nos dados de reconciliação recebidos e na chave bruta, ou

em que o primeiro nó de rede é configurado adicionalmente para

- obter a chave compartilhada e dados de reconciliação a partir da chave bruta,

- enviar os dados de reconciliação ao segundo nó de rede.

13. PRIMEIRO NÓ DE REDE, de acordo com qualquer uma das reivindicações anteriores, caracterizado pelas entradas da matriz compartilhada ($A$) serem polinômios, e em que o polinômio de redução é $x^n + 1$, ou $x^n - 1$, ou $\frac{x^n-1}{x-1}$.

14. PRIMEIRO NÓ DE REDE, de acordo com qualquer uma das reivindicações anteriores, caracterizado pelo circuito de processador ser configurado para computar uma matriz compartilhada expandida para matriz compartilhada (A), substituindo-se as entradas de polinômio de grau maior que 0 por uma matriz circulante, em que a dita matriz expandida compartilhada tem números inteiros módulo o primeiro módulo ($q$) como entradas, e um número de colunas e fileiras igual ao número de colunas e fileiras da matriz compartilhada ($A$) vezes o grau do polinômio de redução.

15. PRIMEIRO NÓ DE REDE, de acordo com qualquer uma das reivindicações anteriores, caracterizado pela adição de ruído ao produto de matriz compreender

- alterar a escala das entradas no produto de matriz para baixo até um segundo módulo ($p$), em que uma entrada escalonada é igual à entrada não escalonada multiplicada com o segundo módulo ($p$), dividido pelo primeiro módulo ($q$) e arredondado para o número inteiro mais próximo, em que o segundo módulo ($p$) é menor que o primeiro módulo ($q$) que computa uma chave bruta como um produto de matriz entre a chave pública recebida do segundo nó e a matriz de chave privada do primeiro nó de rede que é módulo o segundo módulo,

ou

- selecionar elementos de ruído para um ou mais ou todos os elementos no produto de matriz a partir de uma

distribuição de ruído e adicionar os elementos de ruído ao um ou mais ou todos os elementos no produto de matriz, por exemplo, em que os elementos de ruído têm coeficientes selecionados a partir de uma distribuição gaussiana com média 0.

16. PRIMEIRO NÓ DE REDE, de acordo com qualquer uma das reivindicações anteriores, caracterizado por

- a matriz de chave privada $(S_I, S_R)$ ser escolhida de maneira uniformemente aleatória a partir de matrizes de chave privada candidatas, e/ou

- coeficientes de elementos na matriz de chave privada serem ligados em valor absoluto por uma ligação, por exemplo, em que a dita ligação é 2, ou em que a ligação é 1, em que a última corresponde ao binário assinado, e/ou

- colunas e/ou fileiras da matriz de chave privada $(S_I, S_R)$ terem um peso hamming fixo ou ligado $(h_s)$.

17. PRIMEIRO NÓ DE REDE, de acordo com qualquer uma das reivindicações anteriores, caracterizado por

- a matriz $(A)$ ter pelo menos uma dimensão igual a $k$, e opcionalmente ser uma matriz quadrada de $k$ por $k$,

- a chave privada do primeiro nó ter dimensões $n$ e $\bar{n}$,

- a chave privada do segundo nó ter dimensões $\bar{m}$ e $n$, em que $\bar{n}$ e $\bar{m}$ são menores que $n$, por exemplo, $\bar{m}$ e $n$ podem ser 1 ou maiores que 1.

18. PRIMEIRO NÓ DE REDE, de acordo com qualquer uma das reivindicações anteriores, caracterizado pelo circuito de processador ser configurado para

- selecionar todos os números inteiros, polinômios e/ou coeficientes de polinômio na matriz compartilhada a partir do grupo compartilhado, ou

- selecionar parte dos números inteiros, polinômios e/ou coeficientes de polinômio na matriz compartilhada a partir do grupo compartilhado, e gerar a parte restante da matriz compartilhada,

- selecionar todos os números inteiros, polinômios e/ou coeficientes de polinômio na matriz compartilhada a partir do grupo compartilhado, e substituir parte dos números inteiros, polinômios e/ou coeficientes de polinômio na matriz compartilhada de acordo com uma função de substituição.

19. MÉTODO DE OPERAÇÃO CRIPTOGRÁFICA ELETRÔNICA (400) PARA UM PRIMEIRO NÓ DE REDE ELETRÔNICA (110), sendo o método caracterizado por compreender

- dispor (410) comunicação digital entre o primeiro nó de rede e um segundo nó de rede,

- obter (420) uma matriz compartilhada ($A$), em que a matriz compartilhada é compartilhada com o segundo nó de rede através da interface de comunicação, entradas na matriz compartilhada $A$ são números inteiros e/ou entradas na matriz compartilhada ($A$) são polinômios, obter a matriz compartilhada ($A$) compreende

- obter um ou mais parâmetros de seleção compartilhados com o segundo nó de rede,

- selecionar números inteiros, polinômios e/ou coeficientes de polinômio a partir de um grupo compartilhado, em que o grupo compartilhado é compartilhado com o segundo nó de rede e compreende múltiplos números inteiros e/ou polinômios para uso em uma entrada da matriz compartilhada, sendo que a seleção é feita de acordo com uma ou mais funções de seleção, em que uma função de seleção atribui um número inteiro, polinômio e/ou coeficiente de polinômio a partir do

grupo compartilhado para uma entrada na matriz compartilhada, em que o número inteiro, polinômio ou coeficiente de polinômio particular atribuído por uma função de seleção é dependente do parâmetro de seleção,

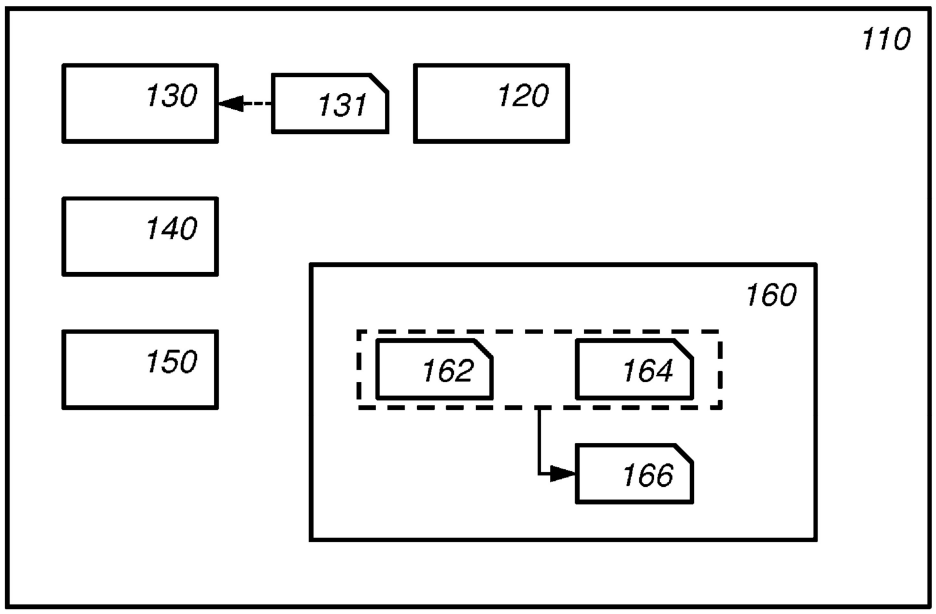
- gerar (430) uma matriz de chave privada ($S_I$), em que entradas na matriz de chave privada são números inteiros e/ou polinômios de número inteiro,

- gerar (440) uma matriz de chave pública ($P_I$)

- computando-se (442) um produto de matriz entre a matriz compartilhada ($A$) e a matriz de chave privada ($S_I$) que obtém um produto de matriz, e adicionando-se ruído (444) às entradas no produto de matriz

- enviar (452) a matriz de chave pública do primeiro nó de rede ao segundo nó de rede.

20. MÍDIA LEGÍVEL POR COMPUTADOR (1000), caracterizada por compreender dados transitórios ou não transitórios (1020) que representam instruções para fazer com que um sistema de processador execute o método, conforme definido na reivindicação 19.


110
100
130
131
120
140
160
150
162
164
166

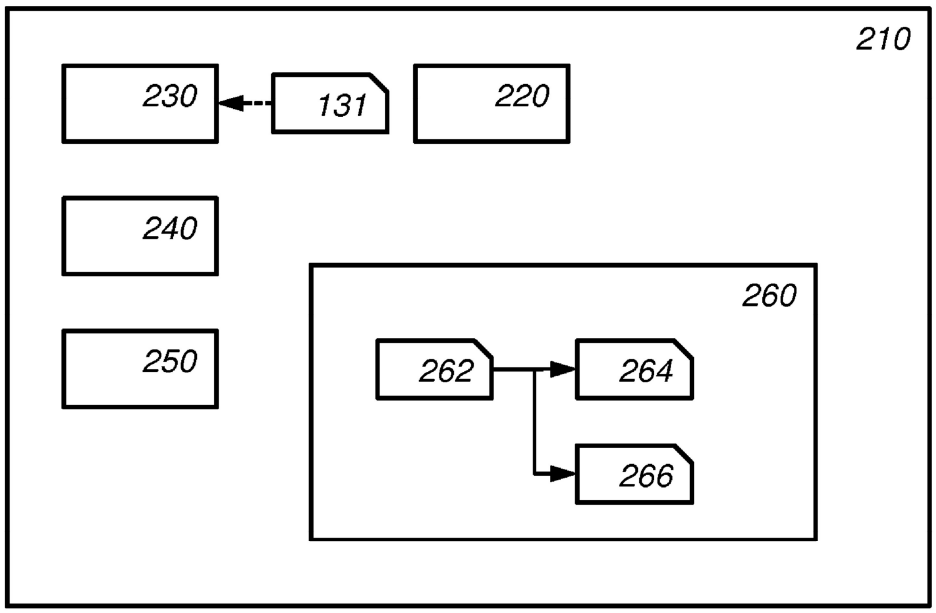

210
230
131
220
240
260
250
262
264
266

Fig. 1

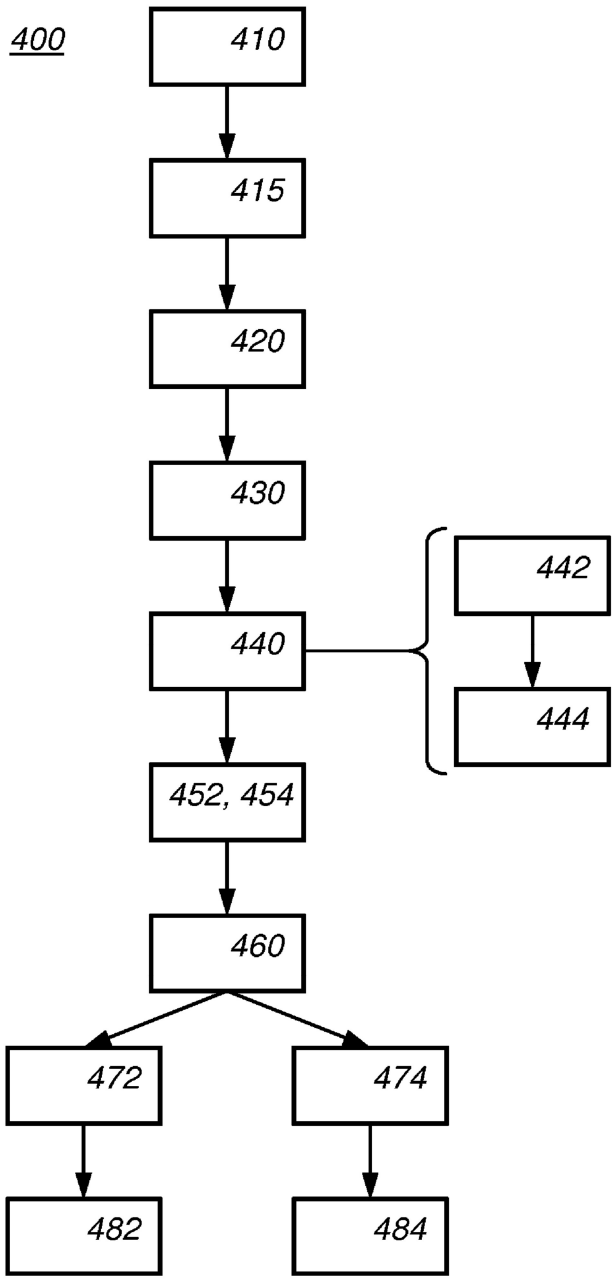



Fig. 2

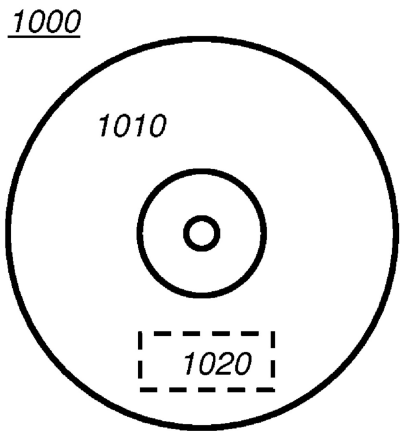



Fig. 3a

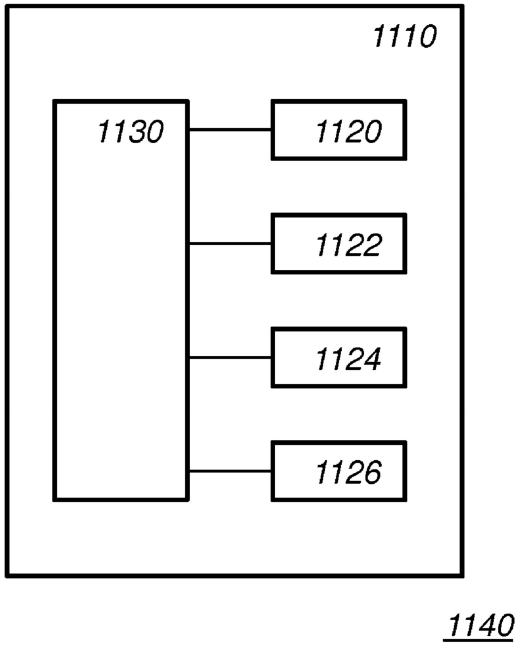



Fig. 3b

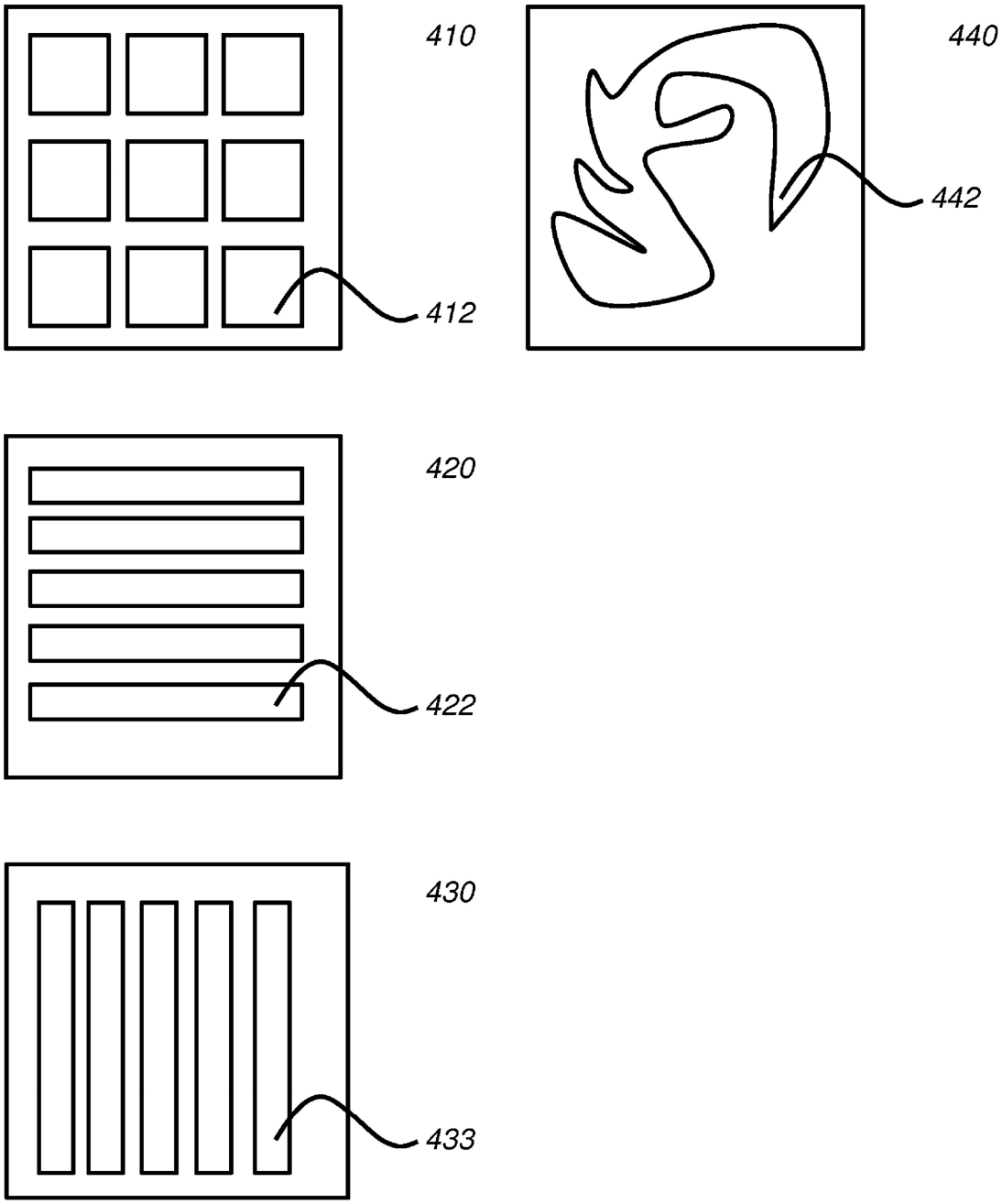



Fig. 4

RESUMO

PRIMEIRO NÓ DE REDE ELETRÔNICA CONFIGURADO PARA UMA OPERAÇÃO CRIPTOGRÁFICA, MÉTODO DE OPERAÇÃO CRIPTOGRÁFICA ELETRÔNICA PARA UM PRIMEIRO NÓ DE REDE ELETRÔNICA E MÍDIA LEGÍVEL POR COMPUTADOR

Trata-se de um nó de rede eletrônica (110) configurado para uma operação criptográfica. O nó de rede obtém uma matriz compartilhada ($A$) selecionando-se números inteiros, polinômios e/ou coeficientes de polinômio a partir de um grupo compartilhado, em que o grupo compartilhado é compartilhado com o segundo nó de rede, em que a seleção é realizada de acordo com uma ou mais funções de seleção.